



UNIVERSITÀ DI PISA

**Facoltà di Ingegneria**

**Corso di Laurea Magistrale in Ingegneria Aerospaziale**

**Tesi di laurea**

**Comfort Termico in ambito  
automobilistico: Studio del transitorio di  
un flusso d'aria fredda in condizioni di  
caldo estremo**

*Candidato*

**Daniele Zhang**

*Relatori*

**Prof. Ing. Giovanni Lombardi**

**Ing. Marco Maganzi**

ANNO ACCADEMICO 2013/2014



*Alla Mia Famiglia ..*

“La vita è come andare in bicicletta: se vuoi stare in equilibrio  
devi muoverti.”

*-Albert Einstein –*



# Indice

<b>Elenco delle figure</b>	8
<b>Introduzione</b>	13
<b>Capitolo 1 - Helyx©</b>	14
<b>1.1 Definizione della cartella di progetto</b>	14
<b>1.2 Struttura generale dei file</b>	16
1.2.1 File della mesh	17
1.2.2 Contorno	17
1.2.3 File di campo	18
<b>1.3 Generazione della Mesh</b>	20
1.3.1 blockMesh	20
1.3.2 snappyHexMesh	22
1.3.3 checkMesh	31
<b>1.4 Impostazione del caso: caseSetup</b>	33
1.4.1 Controlli globali	38
1.4.2 Inizializzazione dei campi	40
1.4.3 Condizioni al contorno	41
<b>1.5 Controllo della soluzione</b>	47
1.5.1 Impostazioni temporali	47
1.5.2 Funzioni	49
<b>1.6 Solutori</b>	53
1.6.1 buoyantBoussinesqSimpleFoam	54
1.6.2 buoyantBoussinesqPimpleFoam	54
1.6.3 Controllo dei solutori	55
1.6.4 Output del solutore	55

<b>1.7 Parallelizzazione</b>	56
1.7.1 decomposePar	56
1.7.2 reconstructParMesh	58
1.7.3 reconstructPar	58
<b>1.8 Visualizzazione</b>	59
<b>1.9 Pre-Processing Avanzato</b>	60
1.9.1 parMapFields	60
<b>Capitolo 2 – Pre-processing</b>	61
<b>2.1 Geometria</b>	61
<b>2.2 Costruzione della mesh</b>	67
<b>Capitolo 3 – Solving</b>	75
<b>3.1 Modello radiativo e solare in Helyx</b>	75
<b>3.2 Caso stazionario</b>	77
3.2.1 ASHRAE 2009	78
3.2.2 I vetri e l'effetto serra	84
3.2.3 Soluzione del carico solare	86
3.2.4 Impostazione del solutore per il caso stazionario	87
<b>3.3 Caso non stazionario</b>	93
3.3.1 Impostazione del solutore per il caso non stazionario	93
<b>Capitolo 4 – Risultati</b>	101
<b>4.1 Risultati del caso stazionario</b>	101
<b>4.2 Risultati del caso non stazionario</b>	109
4.2.1 Risultati sonde	111
4.2.2 Ultimo istante	125
<b>Conclusioni</b>	129
<b>Bibliografia</b>	130



# Elenco delle Figure

## Capitolo 1

Fig.1. 1 Esempio di struttura della cartella per un caso Seriale .....	15
Fig.1. 2 Esempio di struttura della cartella per un caso Parallelo .....	16
Fig.1. 3 Gerarchia delle condizioni al contorno.....	18
Fig.1. 4 Esempio di file di campo: Campo di velocità iniziale .....	19
Fig.1. 5 Dimensioni in OpenFOAM .....	19
Fig.1. 6 Definizione di blockMesh.....	20
Fig.1. 7 Esempio di blockMeshDict.....	21
Fig.1. 8 Esempio di snappyHexMeshDict .....	24
Fig.1. 9 Livelli di raffinamento .....	27
Fig.1. 10 Strati di celle .....	29
Fig.1. 11 Angoli per la definizione della Skewness.....	31
Fig.1. 12 Report checkMesh .....	33
Fig.1. 13 Esempio di caseSetupDict .....	37
Fig.1. 14 Flusso di calore alla parete con condizione al contorno sulla temperatura convettiva .....	43
Fig.1. 15 Esempio di condizione al contorno sulla temperatura della parete con flusso di calore convettivo.....	45
Fig.1. 16 Esempio di condizione al contorno sulla radiazione alla parete.....	46
Fig.1. 17 Esempio di condizione al contorno sulla temperatura per avere la corretta risoluzione della radiazione .....	47
Fig.1. 18 Esempio di file controlDict.....	48
Fig.1. 19 Esempio di funzione volProbes .....	50
Fig.1. 20 Esempio di function fieldWrite .....	51
Fig.1. 21 Esempio di funzione courant .....	51
Fig.1. 22 Esempio di funzione radiation .....	52
Fig.1. 23 Esempio di funzione solar .....	52
Fig.1. 24 Esempio di Output per un solutore SIMPLE.....	55
Fig.1. 25 Esempio di output per un solutore PIMPLE a singola fase .....	56
Fig.1. 26 Esempio di file decomposeParDict.....	57
Fig.1. 29 Finestra tipica di ParaView .....	59

## Capitolo 2

Fig.2. 1 Porte, Plancia. Tappeti, Tunnel, Sedili e Bocchette .....	62
Fig.2. 2 Porte, Plancia. Tappeti, Tunnel, Sedili e Bocchette (altra prospettiva).....	62
Fig.2. 3 Parabrezza, Finestrini, Lunotto e Tetto.....	63
Fig.2. 4 Parabrezza, Finestrini, Lunotto e Tetto (altra prospettiva) .....	63
Fig.2. 5 Manichini del Guidatore e del Passeggero .....	64
Fig.2. 6 Bocchette con Inlet, ed i box Outlet .....	64
Fig.2. 7 Esempio di dettagli da pulire sulle superfici rivolte verso il dominio .....	66
Fig.2. 8 Plancia prima della pulizia.....	66
Fig.2. 9 Plancia dopo la pulizia.....	67



Fig.2. 10 Schermata iniziale di Helyx .....	68
Fig.2. 11 Importazione della geometria *.stl .....	68
Fig.2. 12 Scalatura della geometria durante l'importazione dei file *.stl .....	69
Fig.2. 13 Ambiente di generazione della griglia .....	69
Fig.2. 14 Raffinamenti superficiali e volumetrici .....	70
Fig.2. 15 Opzioni avanzate .....	72
Fig.2. 16 Opzioni di Wrapping .....	72
Fig.2. 17 Posizionamento del Material Point .....	73
Fig.2. 18 Visualizzazione della mesh e le sue caratteristiche. ....	73
<b>Capitolo 3</b>	
Fig.3. 1 Proprietà radiative nella sotto-sezione constant del file caseSetupDict .....	76
Fig.3. 2 Esempio di funzione solarRadiation .....	77
Fig.3. 3 Dati astronomici approssimati per il giorno 21 di ogni mese .....	79
Fig.3. 4 Angoli solari per superfici verticali e orizzontali .....	80
Fig.3. 5 Orientazioni della superficie e corrispondente angolo di azimut superficiale, misurato da sud .....	82
Fig.3. 6 Riflettività di vari tipo di terreno .....	84
Fig.3. 7 Spettro del carico solare sulla superficie terrestre .....	85
Fig.3. 8 Variazione delle proprietà radiative dei vetri con l'angolo di incidenza dei raggi solari: A per vetri a doppia resistenza, B per vetri normali e C per vetri termo-assorbenti .....	86
Fig.3. 9 Impostazione dei parametri globali .....	87
Fig.3. 10 Esempio di applicazione delle condizioni al contorno sulla GUI.....	90
Fig.3. 11 Condizioni al contorno per il Parabrezza.....	91
Fig.3. 12 Impostazione degli schemi di discretizzazione della soluzione .....	91
Fig.3. 13 Impostazioni dei controlli Run-Time.....	92
Fig.3. 14 Inizializzazione delle variabili .....	92
Fig.3. 15 Avvio della simulazione .....	93
Fig.3. 16 parametri globali per il caso non stazionario .....	94
Fig.3. 17 Impostazione del materiale .....	94
Fig.3. 18 Esempio di condizione al contorno di una bocchetta .....	96
Fig.3. 19 Esempio di condizione al contorno di outlet. ....	96
Fig.3. 20 impostazioni Run-time per il caso non stazionario.....	98
Fig.3. 21 sezione function in controlDict per impostare fieldWrite .....	99
Fig.3. 22 Esempio di script durante l'esecuzione di parMapFields .....	100
Fig.3. 23 Finestra Solver per il caso non stazionario .....	100
<b>Capitolo 4</b>	
Fig.4. 1 Andamento dei residui .....	101
Fig.4. 2 Vista frontale della visualizzazione .....	102
Fig.4. 3 Vista dall'alto della visualizzazione .....	102
Fig.4. 4 Vista 1 dei manichini .....	103
Fig.4. 5 Vista 2 dei manichini .....	104
Fig.4. 6 Sezione verticale che interseca gambe e braccia .....	104
Fig.4. 7 Sezione orizzontale altezza testa .....	105

Fig.4. 8 Sezione orizzontale altezza busto .....	105
Fig.4. 9 Sezione orizzontale altezza piedi .....	106
Fig.4. 10 Posizioni delle sonde .....	108
Fig.4. 11 Sezione orizzontale del campo di temperature all'altezza delle bocchette (t=1s) .....	109
Fig.4. 12 Sezione orizzontale del campi di velocità (modulo) all'altezza delle bocchette (t=1s) .....	110
Fig.4. 13 Sezione verticale del campo di temperatura in corrispondenza delle bocchette posteriori (t=1s) .....	110
Fig.4. 14 Sezione verticale del campo di velocità (modulo) in corrispondenza delle bocchette posteriori (t=1s) .....	111
Fig.4. 15 Sonda posizionata sul piede sinistro del passeggero .....	112
Fig.4. 16 Risultati della sonda posta sul piede sinistro del passeggero .....	112
Fig.4. 17 Sonda posizionata sulla mano sinistra del passeggero .....	113
Fig.4. 18 Risultati della sonda posta sulla mano sinistra del passeggero .....	113
Fig.4. 19 Sonda posizionata sulla coscia destra del passeggero .....	114
Fig.4. 20 Risultati della sonda posizionata sulla coscia destra del passeggero .....	114
Fig.4. 21 Sonda posizionata sulla fronte del passeggero .....	115
Fig.4. 22 Risultati della sonda posizionata sulla fronte del passeggero .....	115
Fig.4. 23 Sonda posizionata sulla spalla destra del passeggero .....	116
Fig.4. 24 Risultati della sonda posta sulla spalla destra del passeggero .....	116
Fig.4. 25 Sonda posizionata sul petto del passeggero .....	117
Fig.4. 26 Risultati della sonda posizionata sul petto del passeggero .....	117
Fig.4. 27 Sona posizionata sull'addome del passeggero .....	118
Fig.4. 28 Risultati della sonda posizionata sull'addome del passeggero .....	118
Fig.4. 29 Sonda posizionata sul piede destro del guidatore .....	119
Fig.4. 30 Risultati della sonda posta sul piede destro del guidatore .....	119
Fig.4. 31 Sonda posizionata posta sulla mano sinistra del guidatore .....	120
Fig.4. 32 Risultati della sonda posta sulla mano sinistra del guidatore .....	120
Fig.4. 33 Sonda posta sulla fronte del guidatore .....	121
Fig.4. 34 Risultati della sonda posta sulla fronte del guidatore .....	121
Fig.4. 35 Sonda posizionata sulla coscia destra del guidatore .....	122
Fig.4. 36 Risultati della sonda posizionata sulla coscia destra del guidatore .....	122
Fig.4. 37 Sonda posizionata sul petto del guidatore .....	123
Fig.4. 38 Risultati della sonda posta sul petto del guidatore .....	123
Fig.4. 39 Sonda posizionata sull'addome del guidatore .....	124
Fig.4. 40 Risultati della sonda posizionata sull'addome del guidatore .....	124
Fig.4. 41 Sezione orizzontale del campo di temperature all'altezza delle bocchette (t=127s) .....	125
Fig.4. 42 Sezione orizzontale per il campo di velocità (modulo) all'altezza delle bocchette (t=127s) .....	126
Fig.4. 43 Sezione verticale del campo di temperatura in corrispondenza delle bocchette posteriori (t=127s) .....	126

Fig.4. 44 Sezione verticale del campo di velocità (modulo) in corrispondenza delle bocchette posteriori (t=127s) .....	127
Fig.4. 45 Temperature sui manichini dopo 127 secondi di simulazione.....	128



# Introduzione

Nel presente lavoro si andrà a studiare il transitorio di un flusso d'aria fredda all'interno dell'abitacolo di un'automobile in condizioni di caldo estremo. Questo studio si colloca nell'ambito del comfort termico, ovvero la predizione delle sensazioni termiche e quindi del benessere psicofisico degli occupanti in un ambiente a certe condizioni. La geometria dell'abitacolo analizzata è quella della Ferrari F151. Le simulazioni termofluidodinamiche verranno effettuate con Helyx, un software per CFD (Computational Fluid Dynamics) basato sulle librerie open source OpenFoam sviluppato da Engys. Un altro degli scopi di questo lavoro è appunto testare le potenzialità di questo nuovo software. Sono stati già effettuati altri lavori nello stesso ambito con precedenti versioni di Helyx e si vogliono testare le nuove aggiunte che gli sviluppatori hanno implementato. Il lavoro si svolge in due fasi distinte: si calcolerà innanzitutto il campo di temperature a regime all'interno dell'abitacolo in condizioni di aria condizionata spenta con carico solare in condizioni di caldo estremo (Dubai, 15 Agosto, mezzogiorno, Temperatura esterna 40°); successivamente, i risultati così ottenuti, verranno utilizzati come condizione iniziale per un caso stavolta con bocchette aperte e aria condizionata al massimo studiandone il transitorio. Nel Capitolo 1 si farà una sintesi delle caratteristiche del software soffermandosi sulle novità introdotte da Engys dalle versioni precedenti e dalle applicazioni utili al nostro caso. Nel Capitolo 2 si effettuerà la discretizzazione della geometria, ovvero creare la mesh, cercando un compromesso tra accuratezza e costi computazionali, utilizzando gli strumenti di Helyx. Una volta creata la mesh, nel Capitolo 3, vengono impostate le due simulazioni cercando di implementare i modelli fisici reali il più possibile. Infine nel Capitolo 4 verranno visualizzati i risultati e studiati i transitori delle temperature in certe posizioni sui manichini e nell'abitacolo traendo le conclusioni.

# Capitolo 1

## Helyx<sup>©</sup>

Helyx è un software prodotto da Engys Ltd. basato sulle librerie CFD di pubblico dominio OpenFOAM (Open Field Operation and Manipulation).

Quest'ultimo è un software prodotto e distribuito gratuitamente sotto licenza GPL dalla compagnia commerciale OpenCFD Ltd. e scritto principalmente in linguaggio C++. Viene fornito con un'estesa libreria di solutori, sempre ampliata, applicabile ad una vasta generalità di problemi.

Essendo un prodotto di pubblico dominio, è possibile ampliare e estendere liberamente le sue attuali funzionalità così come creare librerie e solutori nuovi.

Engys ha così sfruttato questa opportunità per crearne una versione propria, aggiungendo delle utili funzionalità e semplificando le impostazioni dei casi con la creazione di un'interfaccia grafica user-friendly. Attualmente siamo arrivati alla versione 2.1.1 con la quale è stato eseguito il presente lavoro. Data la vastità del manuale di Helyx, ci limiteremo a descrivere gli aspetti principali del software e limitando i dettagli ai comandi utilizzati. Si rimanda a [\*] per la descrizione completa e approfondita.

### 1.1 Definizione della cartella di progetto

Ogni caso CFD creato da Helyx è definito da una serie di file di testo e cartelle collocati dentro la cartella di progetto principale. Ogni file e cartella ha un nome, locazione e funzione ben definiti, come mostrato in figura (Fig.1.1) per un caso Seriale. La cartella temporale 0 contiene i file della mesh (all'interno della sottocartella polyMesh) e i file sulle condizioni iniziali dei campi e sulle condizioni al contorno. Una volta che la simulazione inizia verranno create nuove cartelle temporali contenenti le corrispondenti soluzioni di campo relative al particolare tempo o numero di

iterazione (ad es. 100, 200, ecc). Ogni cartella temporale contiene inoltre una cartella *uniform* con un file che descrive l'andamento temporale *time*.

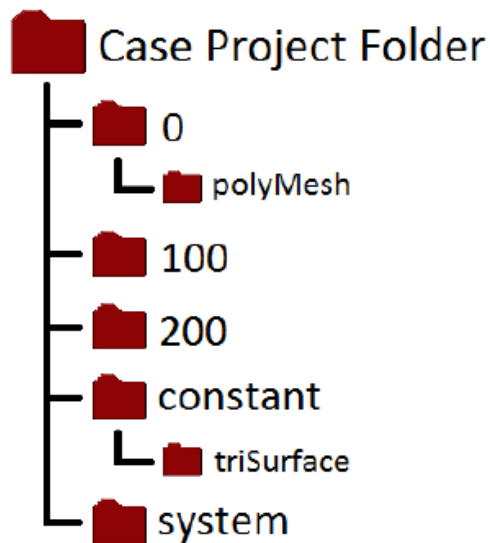


Fig.1. 1 Esempio di struttura della cartella per un caso Seriale

La cartella *system* contiene un numero di file di testo che controllano le applicazioni necessarie per creare la mesh, inizializzare il caso, definire i modelli fisici e le condizioni al contorno, e monitorare la soluzione. Questi file di testo, meglio noti come file di *dictionary*, seguono un particolare formato e nomenclatura che vengono interpretati dalle utility e i solutori durante la simulazione.

La cartella *constant* contiene i file di dictionary che controllano i modelli di turbolenza, le proprietà dei materiali, i mezzi porosi, le componenti rotative (MFR), la radiazione, il vettore di gravità, e altre proprietà. Questa cartella contiene inoltre i file della geometria da meshare in formato .stl collocati all'interno di *triSurface*.

Per un caso eseguito in parallelo invece i dati vengono splittati e salvati nelle varie sottocartelle *processori* come mostrato in (Fig.2.2). Anche se non mostrato in figura la struttura all'interno della cartella *processor0* è la stessa per ogni altra cartella processore. Da notare che il numero delle cartelle processori deve essere lo stesso del numero di CPU utilizzati per l'esecuzione della simulazione (nell'esempio 3 CPU).

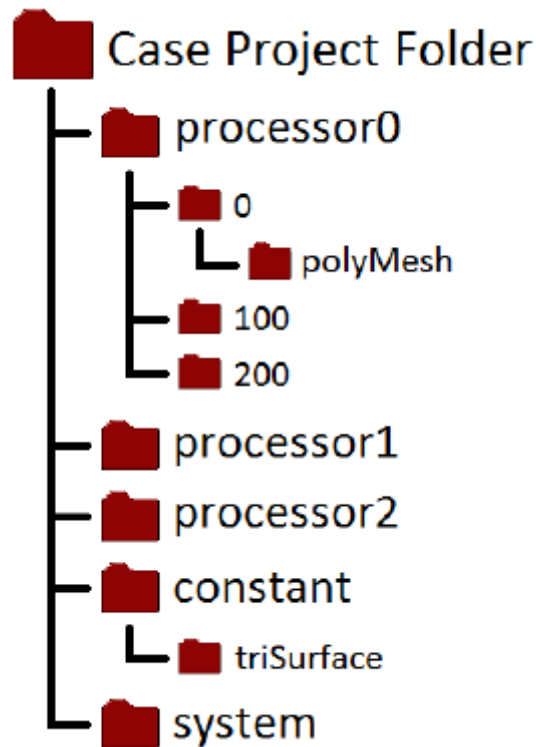


Fig.1. 2 Esempio di struttura della cartella per un caso Parallelo

## 1.2 Struttura generale dei File

Il formato base di un file Helyx è quello usato in OpenFOAM e, quindi, segue i principi del codice C++: le linee non hanno un particolare significato a parte quelle che iniziano con `//` o che sono racchiuse tra `/*` e `*/`, che sono righe di commento e quindi non lette durante l'esecuzione. I documenti hanno una forma libera, senza dare un significato particolare a colonne senza la necessità di indicare la continuazione tra più linee.

Il file classico di OpenFOAM inizia con il nome che lo definisce, *FoamFile*, e poi, tra parentesi graffe, vi è un elenco che riporta le informazioni base come la versione, il formato, la classe, la cartella in cui è posto e la sua natura.

Dopo questa intestazione, si trova la descrizione dei file con tutte le informazioni (inserite tramite liste di dati e organizzate per comunione di contenuto con parentesi tonde) che variano a seconda di quale file si sta considerando.



### 1.2.1 File della Mesh

In OpenFOAM una griglia è definita da un insieme di punti, facce e celle. Ogni cella consiste in un gruppo di facce, mentre ogni faccia a sua volta è composta da un gruppo di punti. Quando una mesh viene creata o importata in OpenFOAM, viene creata la cartella *polyMesh* contenente i seguenti file di mesh:

- *points* è una lista di vettori che definiscono i vertici di ogni cella nella mesh
- *faces* è una lista di facce, dove ogni faccia è definita dalla lista di vertici definita dal file *points*. Le facce possono essere sia interne che al contorno a seconda della loro locazione
- *owner* è una lista di etichette che definiscono quali facce appartengono a quali celle, correlata direttamente ai file *points* e *faces*
- *neighbour* è una lista di etichette che definiscono le celle nelle quali vengono condivise le stesse facce interne
- *boundary* è una lista delle superfici di contorno.

### 1.2.2 Contorno

La definizione delle condizioni al contorno in OpenFOAM segue una gerarchia che può essere divisa in due livelli, come mostrato in (Fig.1.3).

Innanzitutto, le superfici fisiche delle patch devono essere identificate nella mesh per definire il tipo di condizione al contorno nel file *polyMesh/boundary*. Questo passaggio è solo geometrico e coinvolge solo i tipi base.

Secondo, per ciascuna delle suddette superfici bisogna specificare le condizioni al contorno su ogni campo (velocità *U*, temperatura *T*, pressione *p*, *k*, *epsilon*, ecc..). Questi valori sono inseriti nei file di campo presenti nella cartella 0 attraverso il *caseSetup*. Questo passaggio coinvolge solo tipi primitivi e classi derivate.

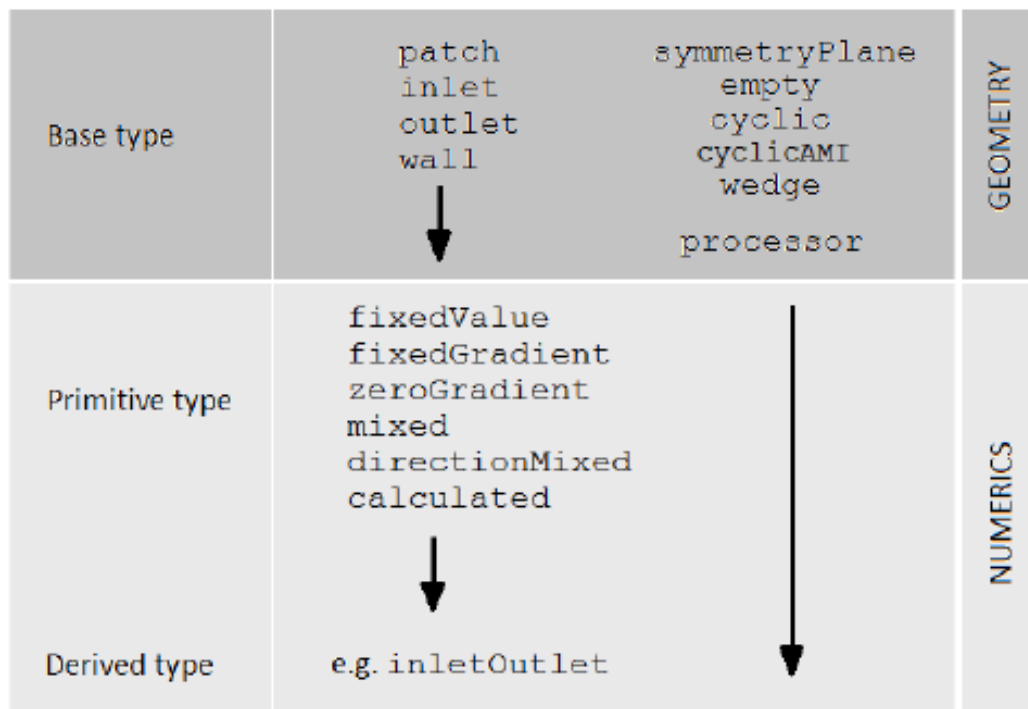


Fig.1. 3 Gerarchia delle condizioni al contorno

### 1.2.3 File di Campo

I file di campo sono collocati nelle cartelle temporali e possono essere sia lette come condizioni iniziali che come risultati del solutore a seconda delle specifiche del utente durante la definizione del caso. In ogni caso i campi sono definiti da tre parole chiave: *dimensions*, *internalField* e *boundaryField*. In Fig.1.4 viene mostrato un esempio di file del campo di velocità come condizione iniziale, tipicamente collocato nella cartella temporale 0.

Dopo l'intestazione FoamFile, la prima parola chiave *dimensions* definisce le unità dimensionali del file campo in questione secondo lo schema in Fig.1.5.

*internalField* può essere descritto come uniforme per assegnare un singolo valore a tutti gli elementi di campo, oppure come non uniforme per cui il valore di ogni singolo elemento viene definito tramite un elenco.

Infine, sotto *boundaryField*, appaiono i nome delle patch, ciascuna con la definizione del tipo (primitivo o derivato) di condizione al contorno e relativo valore.

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "l2000";
    object       U;
}

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    Inlet
    {
        type      fixedValue;
        value      uniform (0.57 0 0);
    }
    Outlet
    {
        type      inletOutlet;
        inletValue uniform (0 0 0);
        value      uniform (0 0 0);
    }
    Wall-Floor
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
    Wall-Sides
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
    Side1
    {
        type      empty;
    }
    Side2
    {
        type      empty;
    }
}

```

Fig.1. 4 Esempio di file di campo: Campo di velocità iniziale

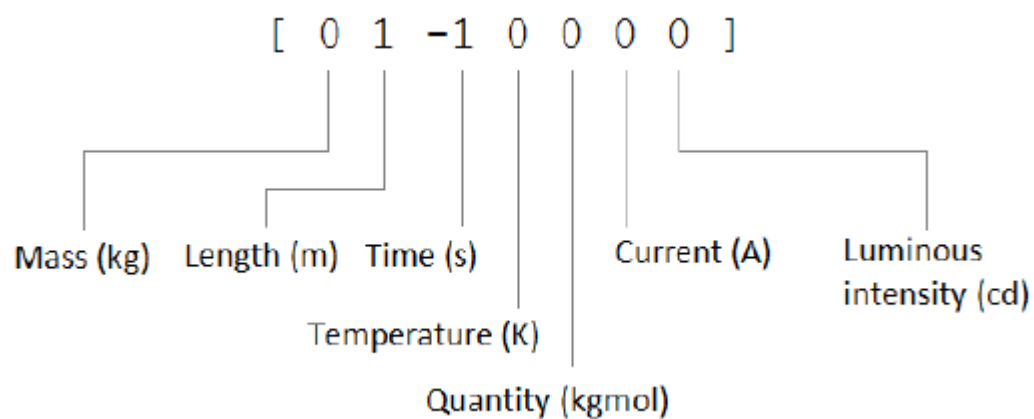


Fig.1. 5 Dimensioni in OpenFOAM

## 1.3 Generazione della Mesh

La generazione della mesh (o griglia) consente di discretizzare la geometria attraverso un reticolo di elementi geometrici più semplici, utile poi per il calcolo delle grandezze fisiche del fluido. OpenFOAM fornisce due applicazioni per la generazione della mesh: *blockMesh* e *snappyHexMesh*. Su quest'ultimo sono state apportate delle modifiche da Engys ed è quello che è stato utilizzato per questo lavoro.

### 1.3.1 blockMesh

L'applicazione *blockMesh* crea griglie strutturate basate su blocchi tridimensionali definite da 8 vertici e 12 spigoli come mostrato in Fig.1.6.

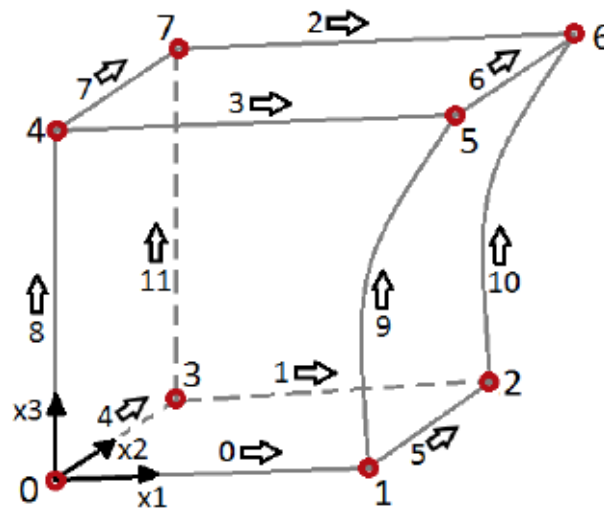


Fig.1. 6 Definizione di *blockMesh*

Il sistema di coordinate locali ( $x_1, x_2, x_3$ ) di ogni blocco segue la regola della mano destra. Una data geometria può essere completamente discretizzata dividendola in blocchi connessi tra loro. Il file che controlla l'applicazione è il *blockMeshDict* nella cartella *system* ed è strutturato come in Fig.1.7.

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
convertToMeters 1.0;

vertices
(
    (0.0 0.0 -0.038)
    (0.038 0.0 -0.038)
    (0.076 0.0 -0.038)
    (0.0 1.09 -0.038)
    (0.038 1.09 -0.038)
    (0.076 1.09 -0.038)
    (0.0 2.18 -0.038)
    (0.038 2.18 -0.038)

```

```

    (0.076 2.18 -0.038)
    (0.0 0.0 0.038)
    (0.038 0.0 0.038)
    (0.076 0.0 0.038)
    (0.0 1.09 0.038)
    (0.038 1.09 0.038)
    (0.076 1.09 0.038)
    (0.0 2.18 0.038)
    (0.038 2.18 0.038)
    (0.076 2.18 0.038)
);

blocks
(
    hex (0 1 4 3 9 10 13 12) (20 400 1) simpleGrading (8 8 1)
    hex (1 2 5 4 10 11 14 13) (20 400 1) simpleGrading (0.125 8 1)
    hex (3 4 7 6 12 13 16 15) (20 400 1) simpleGrading (8 0.125 1)
    hex (4 5 8 7 13 14 17 16) (20 400 1) simpleGrading (0.125 0.125 1)
);

patches
(
    wall ffmaxy
    (
        (6 15 16 7)
        (7 16 17 8)
    )
    wall ffmxy
    (
        (0 1 10 9)
        (1 2 11 10)
    )
    wall cold
    (
        (3 12 15 6)
        (0 9 12 3)
    )
    wall hot
    (
        (2 5 14 11)
        (5 8 17 14)
    )
    empty frontAndBack
    (
        (0 3 4 1)
        (1 4 5 2)
        (3 6 7 4)
        (4 7 8 5)
        (9 10 13 12)
        (10 11 14 13)
        (12 13 16 15)
        (13 14 17 16)
    )
);

edges();
mergePatchPairs();

```

Fig.1. 7 Esempio di blockMeshDict

- *convertToMeters* – serve a scalare le coordinate dei vertici
- *vertices* – in questa sezione vengono specificate, per ciascun vertice che definisce i blocchi, le coordinate  $x_1, x_2$  e  $x_3$ . Ciascun vertice elencato nel file viene numerato con un indice in ordine crescente partendo da 0.
- *blocks* – in questa sezione ciascun blocco e la corrispondente griglia vengono costruiti. Nelle prime parentesi tonde sono definiti i vertici, mentre nelle seconde, in questo caso, è indicato che 14 celle saranno in direzione  $x_1, x_2$  e  $x_3$  rispettivamente. Come ultima cosa, viene impostato il mesh grading, utile se si vuole infittire progressivamente lungo una direzione.
- *edges* – in questa sezione vengono specificati, se presenti, gli spigoli curvi
- *patches* – in questa sezione vengono definite le parti che verranno discretizzate. Viene indicato il tipo di patch, il suo nome e poi, tra parentesi tonde, la lista dei vertici consecutivi che definiscono le facce della patch.
- *mergePatchPairs* – in questa sezione si possono unire più blocchi (già definiti nella sezione *blocks*) per formare un'unica griglia.

Si rimanda a [\*] per ulteriori funzioni e approfondimenti.

### 1.3.2 snappyHexMesh

L'applicazione *snappyHexMesh* consente di creare una griglia tridimensionale, costituita da elementi esaedrici, infittita e modellizzata attorno ad una data geometria. L'idea di base è quella di adattare iterativamente ed automaticamente la griglia del dominio al solido fornito ed eventualmente raffinare la griglia inserendo degli strati di celle più fitti sulle superfici.

Engys ha migliorato quest'applicazione apportando le seguenti modifiche (aggiornato alla versione 2.1 utilizzato in questo lavoro):

- possibilità di esecuzione in parallelo
- mesh dominata da esaedri
- robustezza e qualità della mesh migliorate
- supporto di file geometrici Nastran (.nas) e stereolitografici (.stl)
- correzione automatica di eventuali imperfezioni o buchi della mesh (crack detection)

- controllo sulla crescita degli strati limite
- possibilità di discretizzare in modo diverso particolari zone del dominio (utile per mezzi porosi, MFR, e sorgenti termiche)
- strumento di wrapping, simile al crack detection, ma che fa un controllo globale sull'uniformità delle superfici usando come discriminante le variazioni di pressione (risolve un flusso standard attorno alla geometria e ne determina il campo di pressione, dopo di che tappa le zone in cui questi valori scendono (o superano) al di sotto di un determinato valore deciso dall'utente).

Per essere eseguito correttamente snappyHexMesh richiede:

1. i file della geometria da discretizzare che vanno collocati in constant/triSurface
2. il dominio di calcolo, generato da blockMesh (non necessario se specificato in snappyHexMeshDict – autoBlockMesh attivato)
3. il file snappyHexMeshDict correttamente compilato e collocato nella cartella system, che ne controlla l'intero processo.

Un esempio di snappyHexMeshDict è riportato in Fig.1.8.

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       snappyHexMeshDict;
}

castellatedMesh true;
snap           true;
addLayers      true;
autoBlockMesh true;
blockData (0.03 0);
crackDetection true;
finalDecomposition scotch;

geometry
{
    geom1.stl
    {
        type triSurfaceMesh;
        name domain1;
        transforms
        (
            {
                type translate; translateVec ( 0.5 0 0 );
            }
        );
    }
    geom2.stl
    {
        type distributedTriSurfaceMesh;
        name domain2;
    }
    box
    {
        type searchableBox;
        min (-1.5 -1.5 -0.4);
        max ( 6.0  1.5 1.5);
    }
}

castellatedMeshControls // this section controls the refinement process
{
    include "$FOAM_CONFIG/snappyMesh.castellated";
    // Explicit feature edge refinement
```

```

features();
// Surface based refinement
refinementSurfaces
{
    domain1
    {
        level (3 4);
        regions
        {
            inlet {level (5 5);}
            outlet {level (5 5);}
        }
        proximityIncrement 1;
    }
    domain2
    {
        level (3 4);
        regions {}
        refineSurfaceBoundary true;
    }
    box
    {
        level (4 4);
        faceType internal;
        cellZoneInside inside;
        faceZone faceZoneBox;
        cellZone cellZoneBox;
    }
}

// Region-wise refinement
refinementRegions
{
    domain
    {
        mode distance;
        levels ((0.01 3) (0.05 2) (0.25 1));
    }
}

// Mesh point selection
locationInMesh (1 0 2);
}

snapControls // this section controls the snapping process
{
    featureEdges {}
    include "$FOAM_CONFIG/snappyMesh.snap";
}

addLayersControls // this section controls layer addition
{
    layers
    {
        "domain1.*"
        {
            nSurfaceLayers 5;
            expansionRatio 1.3;
            fch 0.001;
        }
    }
}

```

```

include "$FOAM_CONFIG/snappyMesh.layers";
}

repatchRegions // this section is optional
{
    movingPatch
    {
        location (2.646913 0.88 0.002539);
        zone movingZone;
        excludePatches (amiRight region0 amiRight region0 slave);
    }
}

relaxed // this section is optional for dynamic mesh cases only
{
    maxNonOrtho 72;
    maxBoundarySkewness 22;
    maxInternalSkewness 7;
    maxConcave 82;
    minArea 1e-14;
    minDeterminant 8e-05;
    minFaceWeight 0.07;
    minVolRatio 0.02;
}

critical // this section is optional for dynamic mesh cases only
{
    minVol 1e-14;
}

include "$FOAM_ETC/dictData/snappyMesh.qualityDict"; // quality metrics
include "$FOAM_ETC/dictData/snappyMesh.misc"; // miscellaneous keywords

```

Fig.1. 8 Esempio di snappyHexMeshDict



Durante la conversione dei file in formato .stl, può accadere che si perda l'informazione sull'unità di misura, per cui spesso una geometria con tutte le misure in millimetri può essere trasformata in metri. Per ovviare a questo problema l'interfaccia grafica di Helyx permette di scalare, lungo le tre coordinate, le dimensioni delle geometrie durante l'importazione dei file .stl.

Nella Tab.1.1 si riportano le varie opzioni dell'applicazione

<b>Opzioni</b>	<b>Descrizione</b>
<i>castellatedMesh</i>	Creare una mesh elaborata
<i>snap</i>	Rompere i punti della mesh di superficie della geometria
<i>layers</i>	Generazione di strati e unione di facce
<i>autoBlockMesh</i>	Crea automaticamente la mesh esaedrica iniziale
<i>blockData</i>	Definizione della spaziatura base della mesh creata con autoBlockMesh
<i>crackDetection</i>	Permette di chiudere eventuali buchi presenti nella mesh, ma è costosa computazionalmente
<i>crackTol</i>	Tolleranza usata per crackDetection
<i>finalDecomposition</i>	Specifica il metodo usato per la decomposizione finale della mesh
<i>debug</i>	Controllo dei dati di output
<i>mergeTolerance</i>	Margine di tolleranza in rapporto alla griglia iniziale

Tab.1. 1 Descrizione della prima parte del file snappyHexMeshDict

*snappyHexMesh* richiede una mesh esaedrica iniziale, o di base, per inizializzare il processo di discretizzazione; tutte le celle di questa mesh devono avere un allungamento pari a circa 1 per scongiurare errori sulla soluzione.

In *blockData*, il primo valore numerico indica la spaziatura della mesh creata con *autoBlockMesh*, il secondo a quale livello di cella deve corrispondere tale spaziatura. Di default, il livello della cella scelto è quello base, ovvero 0.

Le varie sezioni che compongono *snappyHexMeshDict* sono:

1. *geometry*, in cui si definiscono tutte le geometrie in corrispondenza delle quali si vuole modellare e infittire la griglia e devono essere menzionate le superfici del solido ed eventuali regioni del dominio, su cui si vuole raffinare la mesh.
2. *castellatedMeshControls*: gestisce il processo di raffinamento. Si specificano quali celle debbano essere infittite e come debbano essere gestite da un punto di vista computazionale.

Le principali opzioni sono:

- *refinementSurfaces*, in cui si definiscono i livelli (*level*) minimi e massimi per i quali la mesh di base sulle superfici di una patch deve essere infittita (Fig.1.9), *featureRefineAngle*, che permette di raffinare la mesh al livello massimo prescritto da *level* su tutti i bordi delle superfici con un angolo (in gradi) superiore al valore prescritto, *regions*, in cui si possono definire i livelli di infittimento della griglia e il *featureRefineAngle* per ogni singola patch che compone l'oggetto in questione e il *proximityIncrement*, che è un valore intero positivo che viene inserito per definire un livello di raffinamento superiore a quello massimo dato in *level*.
  - *refinementRegions*, in cui si definiscono il mode, che può essere *inside* (dentro il volume), *outside* (fuori dal volume) o *distance* (ad una certa distanza dalla superficie) e i *levels*, in cui si vanno a specificare una serie di distanze e livelli per raffinare lontano dalla superficie. Più si è vicini alla superficie e più il raffinamento sarà alto per caratterizzare al meglio la geometria.
3. *snapControl*: completata la fase *castellatedMeshControls*, la superficie del solito risulta dentellata e, quindi, la mesh deve essere smussata e aggiustata. Il procedimento è iterativo e consiste in

- spostare i vertici della griglia sul volume \*.stl
- risolvere i problemi causati nella griglia da tale movimento
- identificare i vertici che possono violare i parametri della qualità della mesh
- ridurre lo spostamento di questi ultimi dal valore iniziale, ripercorrendo quindi i passi precedenti

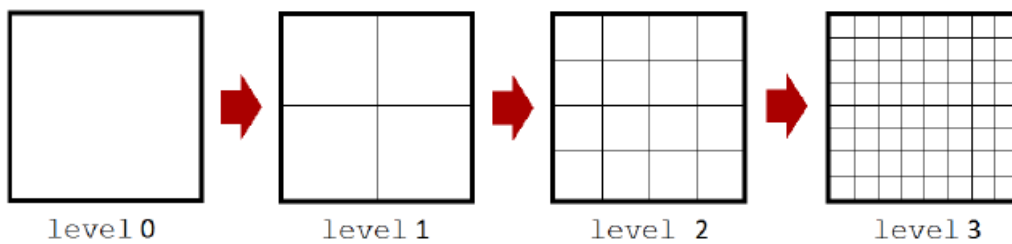


Fig.1. 1 Livelli di raffinamento

4. *addLayersControls*: dopo aver completato la fase di *snapping*, ci sono due stadi per la generazione dei *layers*. Nel primo stadio le facce di contorno vengono unite per aumentare la qualità della mesh, mentre, nel secondo stadio, vengono fatti crescere gli strati attorno alle patch, o ad un gruppo di patch. La procedura avanza nel modo seguente:

- la griglia viene spostata di uno spessore specificato nella direzione normale alla superficie
- si risolvono eventuali problemi nella griglia dati da tale spostamento
- si controlla se è una mesh valida, altrimenti si riduce lo spessore di spostamento e si riparte dal punto sopra; se nessuno spessore è soddisfacente, non si inserisce nessuno strato
- se i criteri di validità sono soddisfatti, si inserisce lo strato di celle
- la griglia viene nuovamente controllata; se non supera il controllo viene rimosso e si riparte dal secondo punto

All'interno della sezione bisogna specificare tutte le patch su cui si vuole inserire uno strato e la quantità di strati

Le opzioni principali di questa sezione sono elencati nella Tab.1.2 con riferimento alla Fig.1.10

<b>Opzione</b>	<b>Descrizione</b>
<i>nSurfaceLayers</i>	Numero di strati da inserire
<i>expansionRatio</i>	Fattore di espansione per lo strato della mesh
<i>finalLayerThickness</i>	E' il rapporto tra lo spessore dello strato piu lontano dalla superficie e la dimensione della mesh superficiale
<i>maxLayerThickness</i>	E' il rapporto tra l'altezza totale degli strati e la dimensione della mesh superficiale
<i>fch</i>	E' l'altezza in metri, del primo strato di celle
<i>minThickness</i>	Spessore minimo delle celle dello strato (se non può essere maggiore non si aggiungono strati)
<i>nRelaxIter</i>	Massimo numero di passi di correzione per riportare la mesh ad uno stato senza errori
<i>maxFaceThicknessRatio</i>	La crescita dello strato si ferma su celle altamente deformate che superano questo valore
<i>maxLayerIter</i>	L'aggiunta di strati si ferma se non viene trovata una mesh senza errori entro questo numero di iterazioni
<i>layerRecovery</i>	Definisce quanto rapidamente viene aggiustato uno strato se questo collassa

Tab.1. 2 Opzioni principali del file *snappyHexMeshDict* -1

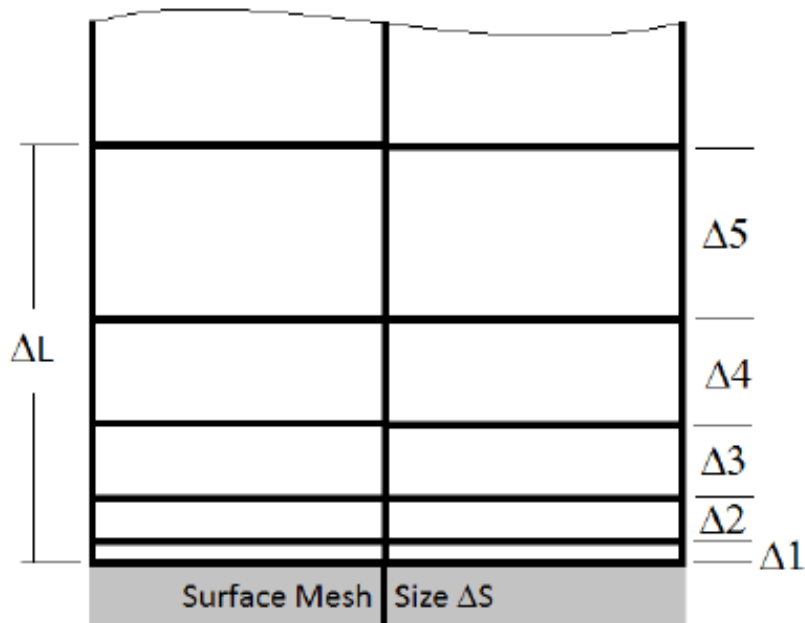


Fig.1. 2 Strati di celle

5. *meshQualityControls*: consente di definire i criteri di validità della mesh, per prevenire eventuali instabilità durante la simulazione. Qualora qualcuno di questi criteri non sia soddisfatto, degli algoritmi di correzione modificano localmente la griglia fino a riportarla entro parametri di qualità accettabili. Si riportano in Tab.1.3 le principali opzioni per questa sezione.

Opzione	Descrizione
<i>maxNonOrtho</i>	Massima non-ortogonalità consentita (porre 180 per disattivare l'opzione)
<i>maxBoundarySkewness</i>	Massima Skewness consentita sui bordi (porla minore di 0 per disattivare l'opzione)
<i>maxInternalSkewness</i>	Massima Skewness consentita internamente (porla minore di 0 per disattivare l'opzione)
<i>minVol</i>	Minimo volume piramidale assoluto (porre -1e30 per disattivare l'opzione)
<i>minArea</i>	Area minima della faccia (porla minore di 0 per disattivare l'opzione)

<i>minTwist</i>	Minima torsione della faccia (porla minore di 0 per disattivare l'opzione)
<i>minVolRatio</i>	Minimo rapporto tra volumi di celle adiacenti (porla minore di 0 per disattivare l'opzione)
<i>maxConcave</i>	Massima concavità consentita (0=diritta,<0=convessa,180=disattivata)
<i>minFlatness</i>	Rapporto tra mini area proiettata e la reale (porla -1 per disattivare l'opzione)
<i>minDeterminant</i>	Determinante minimo della cella normalizzata (porla minore di 0 per disattivare l'opzione)
<i>minTriangleTwist</i>	Deve essere maggiore di 0 per la comparibilità con Fluent
<i>nSmoothScale</i>	Numero di iterazioni per distribuzione dell'errore
<i>errorReduction</i>	Quantità di cui scalare lo spostamento durante ciascuna iterazione di riduzione dell'errore

Tab.1. 3 Opzioni principali del file snappyHexMesh - 2

La Skewness è una caratteristica molto importante per la qualità della mesh. Un'alta Skewness compromette l'accuratezza della mesh: per le celle esaedriche e piramidali non dovrebbe essere maggiore di 0.85. La Skewness può essere definita nei seguenti modi:

- per triangoli e tetraedri:  $Sk = \frac{D_0 - D_r}{D_0}$

dove:

Do=dimensione ottimale della cella

Dr=dimensione reale della cella

- per qualsiasi forma e quasi sempre nel caso di piramidi e prismi:

$$Sk_{quadrato} = \max\left[\frac{\theta_{max} - 90}{90}, \frac{90 - \theta_{min}}{90}\right]$$

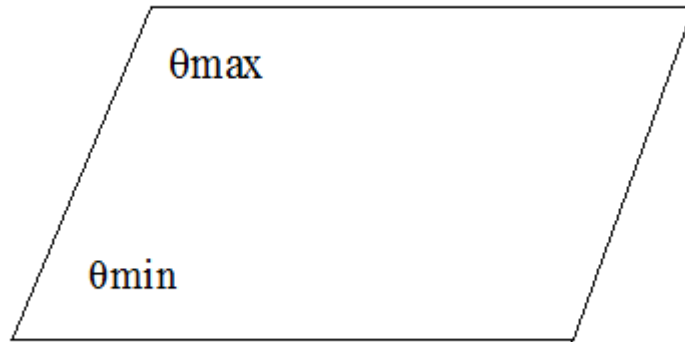


Fig.1. 3 Angoli per la definizione della Skewness

Engys ha implementato l'applicazione *snappysetup* per costruire automaticamente lo scheletro del file *snappuHexMeshDict*, da riempire a mano o con l'interfaccia grafica. Questo permette di risparmiare molto tempo e di ovviare alla possibilità di errori di struttura.

### 1.3.3 checkMesh

Quest'applicazione effettua una serie di test topologici e geometrici per verificare la validità della mesh.

L'output è un report (Fig.1.11) diviso in 3 parti:

1. report delle statistiche di base della mesh
2. controllo sulla topologia della mesh
3. controllo sulla qualità geometrica della mesh

La prima parte presenta il numero totale delle celle, delle facce, delle patch, delle zone e il numero totale di celle divise per tipologia.

Nella seconda parte vengono effettuati test topologici sui bordi, punti, sulle facce e sui loro vertici e vengono contate e controllate le regioni presenti.

Se l'applicazione è eseguita con l'opzione *allTopology*, allora verranno effettuati test anche sulla topologia delle patch, ovvero sulle loro dimensioni e connessioni con altre patch.

L'ultima parte del report riporta i controlli sulle dimensioni della mesh, su eventuali aperture sui bordi, sull'allungamento delle celle e sul loro volume, sull'area delle facce e sulla loro ortogonalità, su volume delle piramidi e sulle Skewness delle facce.

Se tutti i test danno esito positivo, si avrà un buon livello di confidenza che la simulazione andrà a buon fine.

Entro certi limiti ragionevoli, è accettabile che qualche test geometrico come la Skewness, l'allungamento e la minima area delle facce fallisca. Invece tutti i test topologici sono critici e, se falliscono, conducono a problemi di stabilità.

```
/*
|   o   o   | OpenFOAM (Engys Edition):
|   o   o   | Version: 2.2 engysEdition-beta
|   o   o   | Web:    http://www.engys.com
|   o   o   |
*/
Build : 2.2 engysEdition-beta-409ee9a603a2f
Exec   : checkMesh
Date   : Jan 01 0000
Time   : 00:00:00
Host   : model
PID    : 21250
Case   : /home/test/exampleMesh

// ***** //

Create polyMesh for time = 0

Time = 0

Mesh stats
points:      11196
faces:       87464
internal faces: 83786
cells:       20046
boundary patches: 7
point zones: 0
face zones:  0
cell zones:  0
Overall number of cells of each type:
hexahedra:   21043
prisms:      2777
wedges:      817
pyramids:    83
tet wedges:  101
tetrahedra:  34
polyhedra:   1171

Checking topology...
Boundary definition OK.
Point usage OK.
Upper triangular ordering OK.
Cell-faceFace connectivity OK.
Face vertices OK.
Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces ...
```



```

Patch      Faces    Points  Surface topology
ceiling Cube.003  256    263    ok (non-closed singly connected)
floor Cube.006   623    504    ok (non-closed singly connected)
human Sphere     1775   1452    ok (non-closed singly connected)
xmax Cube.002   256    266    ok (non-closed singly connected)
xmin Cube.001   256    265    ok (non-closed singly connected)
ymax Cube.004   256    266    ok (non-closed singly connected)
ymin Cube.005   256    270    ok (non-closed singly connected)

Checking geometry...
Overall domain bounding box (-1.0 -1.0 -1.0) (1.0 1.0 1.0)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (2.668000218e-17 -4.843463437e-17 5.412049151e-17) OK.
Max cell openness = 3.720816407e-16 OK.
Max aspect ratio = 7.638021671 OK.
Minimum face area = 1.6762e-05. Maximum face area = 0.02712. Face area magnitudes OK.
Min volume = 7.86e-07. Max volume = 0.00285. Total volume = 7.8009. Cell volumes OK.
Mesh non-orthogonality Max: 66.35930358 average: 9.798338367
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 3.895105727 OK.
Coupled point location match (average 0) OK.

Mesh OK.

```

Fig.1. 4 Report *checkMesh*

Tutti i controlli falliti vengono scritti in *polyMesh/sets* in modo da poterli visualizzare successivamente.

Una mesh senza errori prodotta con *snappyHexMesh*, può risultare con degli errori di Skewness quando si esegue *checkMesh*. Questo perché sono usate delle tolleranze dell'errore sulla Skewness leggermente diverse tra le due applicazioni. Per superare questa limitazione, anche se non ha conseguenze sulla stabilità della soluzione, Engys ha sviluppato l'applicazione *snappyCheckMesh*.

## 1.4 Impostazione del caso: *caseSetup*

Per impostare un caso si usufruisce dell'applicazione *caseSetup*. Engys ha implementato quest'applicazione con l'interfaccia grafica in modo da inizializzare un caso, scrivere le condizioni al contorno in modo più veloce e automatico rispetto alla release standard di OpenFOAM.

Gli obiettivi del processo di automatizzazione sono: definire lo stato della soluzione, creare i file *fvSolution* e *fvScheme* nella cartella *system*, definire il file *controlDict*, creare i file nella cartella *constant*, definire l'inizializzazione dei campi e le condizioni al contorno. Per essere eseguita, l'applicazione richiede che siano presenti

caseSetupDict e controlDict, nella cartella system e la cartella polymesh (la mesh effettuata precedentemente). Un esempio di file caseSetupDict è mostrato in Fig.1.13.

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       caseSetupDict;
}

regions (region0);

region0
{
    modificationSwitches // this section defines dictionary switches
    {
        resetInternalFields true;
        resetBoundaryFields true;
        resetSystemDicts   true;
        resetConstDicts    true;
        resetBoundaryMesh  true;
        deleteUnusedFields false;
        reuseExistingDicts false;
    }

    boundaryMesh // this section defines base boundary types in the mesh
    {
        partialNamed
        {
            wall {type wall;}
            inlet {type inlet;}
            outlet {type outlet;}
        }
        exactNamed
        {
```

```
            interface side1
            {
                type          cyclicAMI;
                matchTolerance 0.0001;
                neighbourPatch interface_side2;
                transform       noOrdering;
            }
            interface_side2
            {
                type          cyclicAIM;
                matchTolerance 0.0001;
                neighbourPatch interface_side1;
                transform       noOrdering;
            }
        }
    }

    global // this is the global section
    {
        state // define solution state
        {
            SIMPLE
            incompressible
            ras
            buoyant
            humidity
            radiation
            CO2
            AoA
        };

        turbulenceModel kOmegaSST; // specify turbulence model

        materials (air); // define material properties
        materialProperties
        {
            air
            {
                rho rho [1 -3 0 0 0 0 0] 1.2;
                TRef TRef [ 0 0 0 1 0 0 0 ] 295;
            }
        }

        fieldMaps // specify field maps
        {
            Tracer    phase;
            DtTracer  D;
        }

        constant // specify constant dictionary settings
        {
            g
            {
                value          ( 0 0 -9.81 );
            }
            transportProperties
            {
                DtTracer  DtTracer  [0 2 -1 0 0 0 0] 2.4e-5;
                SctTracer SctTracer [0 0 0 0 0 0 0] 0.7;
            }
        }
    }
}
```

```

}
system // specify system dictionary settings
{
    controlDict
    {
        startFrom      startTime;
        startTime       0;
        stopAt          endTime;
        endTime         1000;
        writeInterval    100;
        purgeWrite       0;
        writeCompression compressed;
    }
    fvSchemes
    {
        divSchemes
        {
            div(phi,Tracer) Gauss limitedLinear 1;
        }
    }
    fvSolution
    {
        solvers
        {
            Tracer {solver smoothSolver; smoother GaussSeidel;
tolerance 1e-6; relTol 0.1;}
        }
        relaxationFactors
        {
            Tracer 0.7;
        }
    }
}

fields // this section controls fields initialisation
{
    U{initialisation{type potentialFlow;}}
    p_rgh{initialisation{type potentialFlow;}}
    T{initialisation{type fixedValue; value uniform 297.2;}}
    k{initialisation{type turbulentIL; I 0.3; L 0.1; Uref 0.18;}}
    omega{initialisation{type turbulentIL; I 0.3; L 0.1; Uref 0.18;}}
    w{initialisation{type relativeHumidity; relativehumidity 0.25;}}
    CO2{initialisation{type fixedValue; value uniform 0.0;}}
    Tracer{initialisation{type fixedValue; value uniform 0.0;}}
}

boundaryConditions // this section defines boundary conditions
{
    regionDefaults
    {
        wall
        {
            T
            {
                type incompressible::turbulentHeatFluxTemperature;
                value uniform 297.2;
                q uniform 0.0;
                heatSource power;
            }
        }
    }
}

```

```

IDefault
{
    type          greyDiffusiveRadiation;
    T              T;
    emissivity     uniform 0.95;
    value          uniform 0;
}
}
}
partialNamed
{
    wall_manikin
    {
        T
        {
            type      incompressible::turbulentHeatFluxTemperature;
            value      uniform 307;
            q          uniform 58.0;
            heatSource  flux;
        }
    }
    inlet
    {
        U
        {
            type      fixedValue;
            value      uniform (0.25 0 0);
        }
        T
        {
            type      fixedValue;
            value      uniform 298;
        }
        k
        {
            type      fixedValue;
            value      uniform 0.001;
        }
        omega
        {
            type      fixedValue;
            value      uniform 0.5;
        }
        w
        {
            type      relativeHumidityInletOutlet;
            relativeHumidity uniform 0.1;
            value      uniform 0;
        }
        CO2
        {
            type      fixedValue;
            value      uniform 0.01;
        }
        Tracer
        {
            type      fixedValue;
            value      uniform 0.01;
        }
    }
}
}

```

```

outlet
{
    p
    {
        type fixedValue;
        value uniform 0;
    }
}
exactNamed
{
    wall_window
    {
        w
        {
            type phaseChangeHumidity;
            filmMass uniform 0.01;
            droplets true;
            contactAngle 100;
        }
    }
}

functions // this section defines function objects
(
    VRI
    {
        type volumeReport;
        fields (U p);
    }
    probes
    {
        type volProbes;
        probeLocations ( (0 0 0) (1 0.5 1) );
        fields (U p T);
    }
    solarRadiation
    {
        type solarRadiation;
        solarIntensity 500;
        solarDirection ( 1 1 0 );
        transmissivity (wall window 0.8);
    }
    YTracer
    {
        type concentrationTransport;
        fieldName Tracer;
        solverOptions {ddtScheme steadyState;}
    }
);
}

include "$FOAM_CONFIG/caseSetupDict.defaults"; // include defaults

```

Fig.1. 5 Esempio di caseSetupDict

Le sezioni in cui è diviso il file sono:

1. *modificationSwitches* (lasciarle tutte settate su true tranne l'ultima)
2. *boundaryMesh* (specifica i tipi base delle condizioni al contorno)
3. *global*
4. *field*
5. *boundaryConditions*
6. *functions*

Queste sezioni devono essere definite, individualmente per ciascuna regione discretizzata inserita all'inizio di *caseSetupDict*, nell'opzione *regions*.

### 1.4.1 Controlli globali

In questa sezione, l'utente specifica:

- lo stato della soluzione (*state*): nella Tab.1.4 si riportano alcune delle possibili opzioni per *state*. A seconda della combinazione di opzioni scelte, i campi, i file e i controllori della soluzione verranno automaticamente definite da *caseSetup* in modo che soddisfino i requisiti del solutore.

Opzione	Descrizione
SIMPLE o PISO o PIMPLE	SIMPLE per soluzioni stazionarie, PISO o PIMPLE per soluzione non stazionarie
<i>Incompressible</i> o <i>compressible</i>	Controllo del modello di densità del fluido: valore costante per incompressibile, equazione di stato per comprimibile
<i>ras</i> o <i>les</i>	Controllo del modello di turbolenza: modello RANS per <i>ras</i> o modello DES/LES per <i>les</i>
<i>buoyant</i>	Controllo del modello di gravità: per il caso incompressibile è usata l'approssimazione di Boussinesq
<i>humidity</i>	Attiva il modello di trasporto dell'umidità, con condensazione ed evaporazione alle pareti. Solo per incompressibile
<i>radiation</i>	Attiva il solutore termico radiativo
CO2	Attiva il modello di trasporto scalare della CO2
AoA	Attiva il modello della <i>mean-age of air</i> , utile per stabilire la qualità dell'aria in un ambiente
<i>smoke</i>	Attiva il modello del trasporto scalare del fumo

Tab.1. 4 Opzioni della sezione *state* in *caseSetupDict*

- il modello di turbolenza (*turbulenceModel*) in Tab.1.5 si riportano alcuni dei modelli di turbolenza implementati da Helyx

<b>Opzione</b>	<b>Descrizione</b>
<i>Laminar</i>	Flusso laminare (RANS o DES/LES)
<i>SpalartAllmaras</i>	Modello di Spalart-Allmaras (RANS o DES/LES)
<i>kOmega</i>	Modello di Wilcox (RANS)
<i>kOmegaSST</i>	Modello di Menter con lo Shear Stress Transport (RANS)
<i>kEpsilon</i>	Modello standard per alti Reynolds (RANS)
<i>RNGkEpsilon</i>	Re-Normalisation Group RNG, alti Reynolds (RANS)
<i>realizableKE</i>	Modello Realizable, alti Reynolds (RANS)
<i>qZeta</i>	Legge di Gibson e Dafa' Alla: bassi Reynolds, stazionario o non stazionario, incomprimibile (RANS)
<i>oneEqEddy</i>	Modello SGS k-equation eddy viscosity (DES/LES)
<i>Smagorinsky</i>	Modello SGS Smagorinsky (DES/LES)
<i>lowReOneEqEddy</i>	Modello SGS l-equation eddy viscosity, bassi Reynolds (DES/LES)

Tab.1. 5 Principali modelli di turbolenza supportati da Helyx

- le proprietà dei materiali (*materialProperties*), in questa sezione si specifica il tipo di fluido utilizzato nella simulazione, le cui proprietà sono contenute in un apposito database e sono: la densità, la viscosità dinamica, la capacità termica specifica, il numero di Prandtl, la conducibilità termica, la pressione di riferimento e il coefficiente di espansione termico. E' possibile inserire manualmente anche materiali diversi da quelli presenti.

L'unico modello di trasporto attualmente presente è quello Newtoniano. Nel caso di flussi incomprimibili, le proprietà dei materiali sono scritte in *constant/transportProperties*, mentre nel caso di flussi comprimibili, sono scritte in *constant/thermophysicalProperties*. Se si utilizza un flusso comprimibile, l'utente deve specificare il modello termofisico (*thermoType*) e le sue impostazioni (*specie*, *thermodynamics* e *transport*), in modo tale da costruire un sistema pressione-temperatura dal quale saranno ottenute tutte le proprietà.

- campi nuovi (*fieldMaps*)
- aggiunte e modifiche alla cartella *constant*, in cui si possono creare o modificare i file contenuti in essa, specificandone le impostazioni individualmente. Si possono definire il vettore forza di gravità, le proprietà di trasporto, le proprietà radiative, le zone porose e le sorgenti volumetriche di calore
- aggiunte e modifiche alla cartella *system*, in particolare ai file *controlDict*, *fvScheme* e *fvSolution*

#### 1.4.2 Inizializzazione di campi

Dopo i controlli globali si trova la parte dedicata all'inizializzazione dei campi. Nella Tab.1.6 di riportano alcuni dei modi di inizializzazione previsti.

Opzione	Descrizione
<i>type default</i>	Approccio default di inizializzazione
<i>type potentialFlow</i>	Soluzione potenziale per U e p con spessori standard dello strato limite
<i>type fixedValue</i>	Specificare uno scalare o un vettore
<i>type boundaryValue</i>	Mettere il valore del campo pari alla media sul bordo
<i>type turbulentIL</i>	Definire valori fissi per l'intensità della turbolenza, per la lunghezza della turbolenza, per <i>k</i> , per $\omega$ , per $\epsilon$ , per <i>nut</i> e <i>nuTilda</i>



<i>type Prandtl</i>	Analogia di Prandtl per l'intensità della turbolenza, per la lunghezza della turbolenza, per $k$ , per $\omega$ , per $\epsilon$ , per $\nu_t$ e $\nu_{tilda}$ . Valido solo se è presente un campo di velocità iniziale o con <i>type potentialFlow</i>
<i>type buoyant</i>	Distribuzione di pressione stratificata in base alla forza di gravità. Solo per flussi comprimibili
<i>type relativeHumidity</i>	Specificare un valore tra 0 e 1 per la miscela aria-acqua per inizializzare il campo di umidità relativa $w$

Tab.1. 6 Modi di inizializzazione dei campi

Per flussi incomprimibili, la viscosità, la pressione e di campi di sforzo sono cinematici, ovvero per unità di densità. Ad esempio, la pressione  $p$  è definita in  $m^2/s^2$ , cioè  $\frac{p}{\rho}$  [Pa per  $Kg/m^3$ ]. Il campo di pressione  $p_{rgh}^{10}$  deve essere inizializzato per tutti i solutori multifase e per tutti quelli con la forza di gravità.

### 1.4.3 Condizioni al contorno

Una delle sezioni principali del file *caseSetupDict* è la parte relativa alle condizioni al contorno. Queste vanno inserite per tutte le patch per cui è stato definito un tipo base di condizione al contorno (*inlet*, *outlet*, *wall*, *patch*) nella sezione *boundaryMesh* e possono essere inserite in tre modi:

- *regionDefault*: ogni patch che ha la specificata condizione al contorno base assumerà le descrizioni descritte in seguito.
- *partialNamed*: tutte le patch contenenti la parola specificata avranno le condizioni al contorno descritte in seguito. Per una stessa patch sovrascrive *regionDefault*.
- *exactNamed*: si inserisce il nome esatto della patch, come descritto in *polyMesh/boundary*, con le relative condizioni al contorno. Per una stessa patch sovrascrive *partialNamed*.

Identificate le patch, vengono scritte le condizioni al contorno primitive o derivate. In seguito si riportano alcune delle condizioni a contorno presenti in Helyx, in particolare quelle utilizzate nelle simulazioni.

### *Inlet*

Per inlet l'utente deve specificare le condizioni al contorno dei seguenti campi:

1. Velocità, come vettore, perpendicolare al bordo dell'inlet, come portata, come distribuzione non uniforme o tempo-variante.
2. Pressione totale, in combinazione con una velocità normale al bordo dell'inlet o con una velocità avente una direzione specifica, tempo-variante in combinazione con una velocità normale al bordo dell'inlet o con una velocità avente una direzione specifica.
3. Turbolenza, in accordo al modello di turbolenza scelto in global, definendo  $k$  e  $\epsilon$  (modello  $k-\epsilon$ ),  $k$  e  $\omega$  (modello  $k-\omega$ ),  $\nu$  Tilda (modello Spalart-Allmaras),  $k$  e  $\epsilon$  (modello  $q-\zeta$ ),  $k$  (modelli LES di tipo one-equation eddy viscosity).
4. Temperatura, come valore fissato, come temperatura totale (casi comprimibili) o come distribuzione non uniforme.
5. Umidità, quando si risolve l'equazione di trasporto del vapore acqueo
6. Radiazione, quando si risolve l'equazione della radiazione
7. Scalari passivi ( $CO_2$ , AoA, fumo), se è attivata la relativa equazione di trasporto come valore fissato o tempo variante.
8. Frazione di volume, quando si usano solutori multifase.

### *Outlet*

Per outlet l'utente deve specificare le condizioni al contorno dei seguenti campi, in accordo a quanto specificato per gli inlet:

1. Velocità, come vettore nel sistema cartesiano globale, perpendicolare al bordo dell'outlet, come portata, come distribuzione non uniforme o tempo-variante.
2. Pressione, come valore fissato o come valore medio.
3. Radiazione, quando si risolve l'equazione della radiazione.

### *Wall*

Per wall l'utente deve specificare le condizioni al contorno per i seguenti campi:

1. Velocità, come vettore o per definire una parete in movimento o rotante.
2. Temperatura, come valore fissato, per definire il flusso di calore alla parete ( $\text{W/m}^2$ ), per definire il carico di calore alla parete (W) o per definire il trasferimento di calore convettivo e la conduzione.
3. Radiazione
4. Umidità

La condizione di non scorrimento è automaticamente generata per tutte le patch che abbiano wall come condizione al contorno. Tuttavia, è possibile cambiare manualmente tale condizione impostandola su slip (condizione di scorrimento).

La condizione al contorno con il trasferimento di calore per convezione è usata per calcolare  $T_{\text{wall}}$ , inclusi gli effetti di conduzione e lo stato termico al di fuori dalla parete (Fig.1.14)

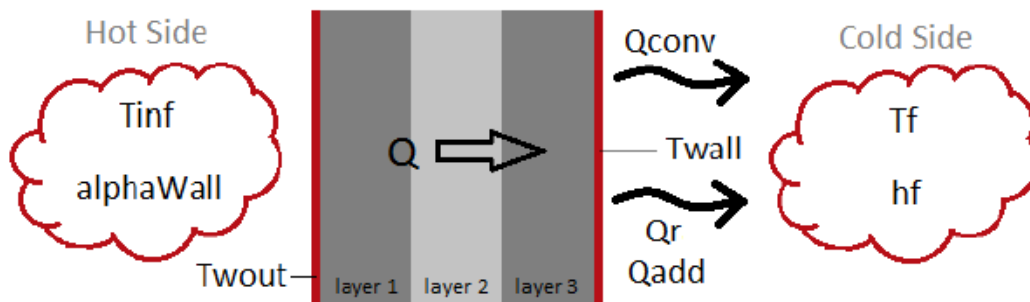


Fig.1. 6 Flusso di calore alla parete con condizione al contorno sulla temperatura convettiva

Ignorando gli effetti di calore latente a causa di cambi di fase, il trasferimento di calore all'interfaccia solido-fluido può essere scritto nel seguente modo:

$$Q = Q_{\text{conv}} + Q_r + Q_{\text{add}} \quad (\text{Eq.1.1})$$

Dove

$Q_{\text{conv}} = hf(T_{\text{wall}} - T_f)$  è il flusso convettivo,  $\text{W/m}^2$

$hf$  è il coefficiente di trasferimento di calore,  $\text{W/K m}^2$

$T_f$  è la temperatura del fluido lontano dalla parete, K

$T_{\text{wall}}$  è la temperatura della parete, K

$Q_r$  è il flusso di calore radiativo totale,  $\text{W/m}^2$

$Q_{\text{add}}$  è un flusso di calore addizionale definito dall'utente  $\text{W/m}^2$

Q può essere scritto anche nel seguente modo:

$$\begin{aligned} Q &= \alpha_{\text{Wall}} \cdot (T_{\text{inf}} - T_{\text{wout}}) \\ &= (\lambda_{\text{layer1}}) \cdot (\Delta T_{\text{layer1}}) / (L_{\text{layer1}}) \\ &= (\lambda_{\text{layer2}}) \cdot (\Delta T_{\text{layer2}}) / (L_{\text{layer2}}) \\ &= (\lambda_{\text{layer3}}) \cdot (\Delta T_{\text{layer3}}) / (L_{\text{layer3}}) \end{aligned} \quad (\text{Eq.1.2})$$

dove:

$\alpha_{\text{Wall}}(T_{\text{inf}} - T_{\text{wout}})$  è il flusso di calore convettivo all'esterno della parete,  $\text{W/m}^2$

$\alpha_{\text{Wall}}$  è il coefficiente di trasferimento di calore all'esterno della parete,  $\text{W/K m}^2$

$T_{\text{inf}}$  è la temperatura ambiente all'esterno della parete, K

$T_{\text{wout}}$  è la temperatura della parete all'esterno, K

$\lambda \Delta T / L$  è il flusso di calore conduttivo attraverso ciascuno strato di materiale di cui è costituita la parete,  $\text{W/m}^2$

$\lambda$  è la conducibilità termica di ciascuno strato di materiale di cui è costituita la parete,  $\text{W/K m}$

$L$  è lo spessore di ciascuno strato di materiale di cui è costituita la parete

$\Delta T$  è la differenza di temperatura lungo la lunghezza dello strato.

In Fig.2.15 si può trovare un esempio di implementazione di questa condizione al contorno. Come si può notare, i valori da impostare sono:

- $\alpha_{\text{Wall}}$  (se l'utente specifica questo valore, allora  $T_{\text{wout}}$  sarà calcolata dall'equazione 3.4, altrimenti  $T_{\text{wout}} = T_{\text{inf}}$ )
- $T_{\text{inf}}$  (l'utente deve sempre definirla)
- $Q_{\text{add}}$  (opzionale)
- $Thickness$  (è la  $L$  dell'Eq.1.2)
- $Lambda$  (è  $\lambda$  dell'Eq.1.2)

```

T
{
    type    incompressible::turbulentConvectiveTemperature;
    // type    compressible::turbulentConvectiveTemperature;

    value uniform 300;
    alphaWall uniform 0.01;
    Tinf uniform 300;
    qadd uniform 50;

    layers
    {
        outside_metal
        {
            thickness uniform 0.003;
            lambda      uniform 16;
        }
        insulation
        {
            thickness uniform 0.05;
            lambda      uniform 0.03;
        }
        inside_metal
        {
            thickness uniform 0.002;
            lambda      uniform 16;
        }
    }
}

```

Fig.1. 7 Esempio di condizione al contorno sulla temperatura della parete con flusso di calore convettivo

La sezione *Layers* è opzionale: se viene specificata, allora  $T_{\text{wall}}$  sarà calcolato da  $\Delta T$  dell'equazione 1.4, altrimenti  $T_{\text{wall}}=T_{\text{wout}}$ . Per quanto riguarda la radiazione, che è presente se è attiva l'equazione dell'energia, in Helyx è implementato solo il modello per corpi grigi, ovvero valgono le seguenti affermazioni:

- Le superfici sono grigie, cioè le proprietà radiative sono indipendenti dalla lunghezza d'onda
- Le superfici sono diffuse, cioè le proprietà radiative sono indipendenti dalla direzione
- Vale la legge di Kirchoff, cioè assorbività=emissività (l'energia assorbita è uguale a quella emessa)

Questo significa che vale la seguente legge:

$$\varepsilon + \sigma + \tau = 1 \quad (\text{Eq.1.3})$$

dove:

$\epsilon$ =emissività ( $0 < \epsilon < 1$ ), definita dall'utente

$\tau$ =trasmissività ( $0 < \tau < 1$ ), definita dall'utente

$\sigma$ =riflettività ( $0 < \sigma < 1$ ), calcolata dall'Eq.2.3

L'emissività è il rapporto tra la radiazione emessa da una superficie e la radiazione emessa da un corpo nero alla stessa temperatura. Il potere emissivo totale ( $\text{W/m}^2$ ) di una superficie è dato dalla legge di Stefan-Boltzmann:

$$E(T) = \epsilon(T)\sigma T^4 \quad (\text{Eq.1.4})$$

dove:

$\sigma = 5.67 \cdot 10^{-8} \text{ W/m}^2\text{K}^4$ =costante di Stefan-Boltzmann

Quando la radiazione colpisce un oggetto, essa viene in parte riflessa, in parte assorbita e in parte trasmessa.  $\tau$  è il rapporto tra la radiazione trasmessa e quella incidente, mentre  $\sigma$  è il rapporto tra la radiazione riflessa e quella incidente. Per superfici opache  $\tau = 0$ , mentre per superfici trasparenti è  $\tau = 1$ . In Fig.1.16 è mostrato un esempio di applicazione della condizione al contorno sulla radiazione.

Come si può notare, i valori da assegnare sono:

- emissivity
- transmissivity
- $T_{\text{inf}}$  che rappresenta la temperatura ambientale esterna (o del cielo) ed è utile definirla per superfici semi-opache o trasparenti, ovvero per quelle superfici che hanno una certa trasmissività. Infatti questa temperatura permette di definire la radiazione diffusa dall'ambiente esterno secondo la legge di Stefan-Boltzmann (Eq.1.4), in cui  $\epsilon$  è pari a 1 (l'ambiente esterno è considerato come un corpo nero) e  $T = T_{\text{inf}}$ .

```
IDefault
{
    type greyDiffusiveRadiation;
    value uniform 0;
    T T;
    emissivity uniform 0.95;
    transmissivity 0;
    Tinf 300;
}
```

Fig.1. 8 Esempio di condizione al contorno sulla radiazione alla parete

Si noti che, quando si vuole implementare la radiazione, tutte le pareti adiabatiche devono essere esplicitamente definite dall'utente per avere dei risultati corretti. Questo si ottiene imponendo flusso di calore nullo, come si può vedere nella Fig.1.17.

```
T
{
    type    incompressible::turbulentHeatFluxTemperature;
    // type  compressible::turbulentHeatFluxTemperature;
    value   uniform 297.2;
    q        uniform 0.0;
    heatSource power;
    // Additional entries required for compressible flows:
    K        basicThermo;
```

Fig.1. 9 Esempio di condizione al contorno sulla temperatura per avere la corretta risoluzione della radiazione

## 1.5 Controllo della soluzione

### 1.5.1 Impostazioni temporali

Il controllo e il monitoraggio della soluzione vengono effettuati tramite il file *controlDict* (Fig.1.18) nella cartella *system*. Questo file consiste in una serie di comandi responsabili del controllo del tempo e degli input/output dei dati durante la simulazione. In fondo al file si trovano le funzioni e i campi da monitorare. Il file può essere diviso in due parti principali:

1. Run-time Controls
2. Function Objects

I controlli Run-time permettono di controllare il tempo e i dati di input/output per la soluzione. Le impostazioni temporali della sezione *time settings*, che devono sempre essere definite dall'utente, sono definite dalle seguenti opzioni:

- *startFrom*, in cui viene indicato il tempo/iterazione di partenza della simulazione. L'utente ha tre possibilità tra cui *startTime*, nel quale bisogna specificare da quale cartella temporale si vuole far iniziare la simulazione
- *stopAt*, in cui viene indicato il tempo/iterazione a cui si vuole fermare la simulazione. L'utente ha quattro possibilità tra cui *endTime*, nel quale bisogna specificare il tempo/iterazione al quale si arresterà la simulazione

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       controlDict;
}

// time settings
startFrom      latestTime;
startTime      0;
stopAt         endTime;
endTime        500;
deltaT         1;

// data writing
writeControl    adjustableRunTime;
writeInterval   100;
purgeWrite      0;
writeFormat     ascii;
writePrecision  8;
writeCompression uncompress;
timeFormat      general;
timePrecision   6;
graphFormat     raw;

// data reading
runTimeModifiable true;

// PIMPLE solvers only
adjustTimeStep  false;
maxCo           1000;
maxDeltaT       1;

functions
(
    inlet_surfaceReport
    {
        type surfaceReport;
        functionObjectLibs ( "libreport.so" );
        region region0;
        logToFile true;
        weighting flux; //area
        fluxName phi;
        homogeneity false;
        fields
        {
            U p k epsilon
        }
    };

    surfaces
    {
        {
            type patch;
            name inlet;
        }
    };
    constraints
    {
        {
            type boundBox;
            min (-1000 -1000 -1000);
            max (1000 1000 1000);
            keepInside true;
        }
    };
}

all volumeReport
{
    type volumeReport;
    functionObjectLibs ( "libreport.so" );
    region region0;
    logToFile true;
    fields
    {
        U p k epsilon
    };
}

radiation
{
    type nonParticipatingRadiation;
    functionObjectLibs ( "libsolverFunctionObjects.so" );
    region region0;
}

solar
{
    type solarRadiation;
    functionObjectLibs ( "libsolverFunctionObjects.so" );
    region region0;
    outputControl timeStep;
    outputInterval 1;
    solarIntensity 1000; //W/m2
    solarDirection (1 1 -1);
    transmissivity
    {
        (cabin_window_cabin_window 0.8)
        (cockpit_front_window_cockpit_front_window 0.8)
        (cockpit_side_window_cockpit_side_window 0.8)
    };
}
}
);

```

Fig.1. 10 Esempio di file controlDict



- *deltaT*, che si riferisce al passo temporale della simulazione (per casi non stazionari, come nel presente lavoro, bisogna specificare un valore appropriato per assicurare una sufficiente risoluzione temporale. Per casi stazionari può essere posto uguale a 1 in modo che agisca come un contatore di iterazioni)
- *adjustTimeStep*, che se attivata permette al solutore di modificare il time step automaticamente, durante la simulazione, soddisfacendo un certo criterio (max *deltaT* e/o *maxCo* numero di Courant) per avere il miglior rapporto velocità/stabilità.

La sezione di *data writing* controlla la frequenza di scrittura e le caratteristiche dei dati di output. Si descrivono le seguenti opzioni:

- *writeControl*, che controlla la frequenza di scrittura dei dati di output. L'utente ha cinque possibilità tra cui *timeStep*, nel quale si specifica ogni quanti passi temporali o iterazioni saranno scritti i dati su disco (*writeInterval*), e *runTime*, nel quale si specifica ogni quanti secondi di simulazione saranno scritti i dati (*writeInterval*).
- *purgeWrite*, che si riferisce al numero di cartelle temporali che saranno salvate. Se, ad esempio è pari a 2, solo le due ultime cartelle temporali saranno salvate e le restanti cancellate. Se posto uguale a 0, saranno mantenute tutte le cartelle.
- *writeFormat*, che controlla il formato dei dati salvati (*ascii* o *binary*)
- *writeCompression*, che controlla il tipo di compressione dei dati (uncompressed o compressed)
- *timeFormat*, che controlla il formato delle cartelle temporali (*general*, *fixed*, o *scientific*)
- *graphFormat*, che controlla il formato dei dati (*raw*, *gnuplot*, *xmgr* o *jplot*)

I controlli Run-time possono essere modificati in ogni momento durante la simulazione se attiva l'opzione *runTimeModifiable*. Le modifiche apportate saranno attive una volta completato il passo temporale/iterazione.

### 1.5.2 Funzioni

Le funzioni sono delle routine dedicate che permettono di eseguire diverse operazioni. Le più importanti sono:

1. Il monitoraggio della soluzione

2. La stabilità della soluzione
3. L'esportazione dei dati della soluzione
4. La modellazione della radiazione
5. Il trasporto scalare passivo

La soluzione può essere monitorata con le seguenti funzioni:

- *volumeReport*, che per ogni passo temporale/iterazione dà il minimo, il massimo, la media pesata sul volume, la deviazione standard dei campi selezionati dall'utente, in una prescritta zona della mesh di volume. I dati vengono scritti in un file all'interno della cartella *log*.
- *surfaceReport*, che, per ogni passo temporale/iterazione, dà i valori mediati con l'area, o con il flusso, dei campi selezionati dall'utente, in una prescritta superficie. Vengono riportati anche il minimo, il massimo e la deviazione standard dei campi selezionati, in aggiunta al flusso netto di volume ( $\text{m}^3/\text{s}$ ), per flussi incomprimibili, o la portata ( $\text{kg/s}$ ), per flussi comprimibili, attraverso la patch definita. Anche in questo caso i dati vengono scritti in un file all'interno della cartella *log*.
- *volProbes*, che per ogni passo temporale/iterazione, dà i valori puntuali dei campi selezionati (Fig.1.19): è una sonda. In *probeLocations* si elencano le coordinate cartesiane in cui si vogliono posizionare le sonde, mentre in *fields* si elencano i campi da monitorare. I risultati sono riportati nella cartella *log*.
- *forces*, che per ogni passo temporale/iterazione, dà le pressioni, le azioni viscosi e momenti agenti sulle patch selezionate dall'utente. I risultati sono salvati nel file *forces.dat* all'interno della nuova cartella chiamata *forces*.

```
functions
(
    PP1
    {
        type volProbes;
        functionObjectLibs ( "libreport.so" );
        region region0;
        probeLocations
        {
            (0 0 1)
            (0 1 0)
        };
        fields (U p);
        outputControl outputTime;
        outputInterval 1; //Optional entry depending on outputControl
    }
);
```

Fig.1. 11 Esempio di funzione *volProbes*

- *liftDrag*, che per ogni passo temporale/iterazione dà i coefficienti della forza aerodinamica e del momento agenti su una patch selezionata dall'utente. I dati sono salvati nella cartella *log*.
- *fieldWrite*, che ogni quanto definito dall'utente, permette di scrivere su disco i campi delle grandezze sempre definite dall'utente. Ciò permette di snellire i file di output, andando a salvare le sole grandezze che interessano. La scrittura normale salva tutti i campi soluzione della simulazione e porta a occupare molto spazio su disco. Un esempio viene mostrato in Fig.1.20. *writeControl* e *writeInterval* definiscono ogni quanto salvare, in *fields* si mettono i campi che interessano all'utente e *spanControl* stabilisce un controllo sulle scritture come per esempio un numero massimo di campi.

```

functions
(
    noiseDumps
    {
        type fieldWrite;
        functionObjectLibs ( "libfieldFunctionObjects.so" );
        region region0;
        writeControl timeStep;
        writeInterval 10;
        spanControl startAndNwrites;

        maxWrites 2000;
        start 11;
        end 1000000;
        fields
        {
            DpDt //any valid volume or surface field
        };
    }
);

```

Fig.1. 12 Esempio di function *fieldWrite*

Il monitoraggio della stabilità della soluzione avviene attraverso le seguenti funzioni:

- *courant*, che dà il valore medio del numero di Courant per ogni passo temporale/iterazione (Fig.1.21). Il numero di Courant mette in relazione la risoluzione temporale con quella spaziale ed è definito nel seguente modo

$$Co = U \frac{\partial t}{\partial x} \quad (\text{Eq.2.5})$$

```

functions
(
    courant
    {
        type courant;
        functionObjectLibs ( "libreport.so" );
        region region0;
        logToFile true;
    }
);

```

Fig.1. 13 Esempio di funzione *courant*

- *fieldBlendingFactor*, che permette di aumentare la robustezza e l'accuratezza di simulazioni DES/LES

Per attivare la radiazione termica, viene utilizzata la funzione *radiation* (Fig.1.22), mentre invece per attivare la radiazione solare, viene usata la funzione *solar* (Fig.1.23).

```
functions
{
    radiation
    {
        type nonParticipatingRadiation;
        functionObjectLibs ( "libsolverFunctionObjects.so" );
        region region0;
        outputControl timeStep;
        outputInterval 1;
    }
};
```

Fig.1. 14 Esempio di funzione radiation

```
functions
{
    solar
    {
        type solarRadiation;
        functionObjectLibs ( "libsolverFunctionObjects.so" );
        region region0;
        sources
        {
            //
            transmissivity
            {
                (cabin window 0.8)
                ("cockpit.*" 0.8)
            }
            //
            outputControl timeStep;
            outputInterval 1;
        }
    }
};
```

Fig.1. 15 Esempio di funzione solar

La sotto-sezione *sources*, che è facoltativa, contiene una lista supplementare di sorgenti solari, di cui vanno definite intensità e direzione. In *transmissivity* si definiscono le patch da trattare come oggetti trasparenti o semi-trasparenti e la trasmissività associata.

Infine la funzione del trasporto scalare passivo viene usata quando si vuole rappresentare la concentrazione di gas, come CO<sub>2</sub> e fumo, e si vuole valutare la qualità dell'aria.

## 1.6 Solutori

Nella Tab.1.7, si riportano alcuni dei solutori presenti in Helyx.

Opzione	Descrizione
<i>simpleFoam</i>	Stazionario, incomprimibile, RANS
<i>pimpleFoam</i>	Non stazionario, incomprimibile, URANS/LES
<i> pisoFoam</i>	Non stazionario incomprimibile, URANS/LES
<i>buoyantBoussinesqSimpleFoam</i>	Stazionario, incomprimibile, RANS, con solutore termico e forza di gravità
<i>buoyantBoussinesqPimpleFoam</i>	Non stazionario, incomprimibile, URANS/LES, con solutore termico e forza di gravità
<i>rhoSimpleFoam</i>	Stazionario, comprimibile, RANS, con solutore termico
<i>rhoPimpleFoam</i>	Non stazionario, comprimibile, URANS, con solutore termico
<i>buoyantSimpleFoam</i>	Stazionario, comprimibile, RANS, con solutore termico e forza di gravità
<i>buoyantPimpleFoam</i>	Non stazionario, comprimibile, URANS, con solutore termico e forza di gravità

Tab.1. 7 Solutori presenti in Helyx

Definito in *caseSetupDict* lo stato della soluzione (state), esiste un unico solutore associato ad esso. Nel presente lavoro si è dapprima utilizzato *buoyantBoussinesqSimpleFoam* per i casi stazionari con radiazione solare e bocchette chiuse, successivamente si è utilizzato *buoyantBoussinesqPimpleFoam* per valutare i casi non stazionari. Perciò verranno approfonditi questi solutori, mentre per gli altri si rimanda al manuale di Helyx.

### 1.6.1 buoyantBoussinesqSimpleFoam

Il solutore *buoyantBoussinesqSimpleFoam* ha le seguenti caratteristiche:

- stato stazionario
- incomprimibile
- non-isotermico (risolve l'equazione dell'energia)
- laminare o turbolento (RANS)
- radiazione termica
- radiazione solare
- umidità
- forza di gravità (approssimazione di Boussinesq)
- sorgenti di calore volumetriche (zone termiche)
- mezzi porosi
- MFR
- Trasporto scalare passivo

### 1.6.2 buoyantBoussinesqPimpleFoam

Il solutore *buoyantBoussinesqPimpleFoam* ha le seguenti caratteristiche:

- stato non-stazionario
- incomprimibile
- non-isotermico (risolve l'equazione dell'energia)
- laminare o turbolento (URANS, DES, LES)
- radiazione termica
- radiazione solare
- umidità
- forza di gravità (approssimazione di Boussinesq)
- sorgenti di calore volumetriche (zone termiche)
- mezzi porosi
- MFR
- Trasporto scalare passivo

### 1.6.3 Controllo dei solutori

Tutti i solutori sono controllati da due file chiamati *fvSolution* e *fvScheme*, che si trovano all'interno della cartella *system*.

Il primo è necessario perché prescrive le equazioni del solutore, le tolleranze e gli algoritmi della soluzione. Il file viene automaticamente creato una volta eseguito il *caseSetup* e i suoi parametri sono direttamente dipendenti dalle scelte fatte nell'apposita sezione del *caseSetupDict*.

Il secondo contiene gli schemi di discretizzazione dei vari operatori matematici presenti nelle equazioni del problema (tempo, gradiente, divergenza, laplaciano, ecc.). Anche questo file viene creato tramite il *caseSetup* con i parametri contenuti nell'apposita sezione all'interno del *caseSetupDict*.

### 1.6.4 Output del solutore

Dei tipici esempi di output di un solutore SIMPLE e un solutore PIMPLE sono mostrati in Fig.1.24 e Fig.1.25.

```
Time = 5
smoothSolver: Solving for Ux, Initial residual = 0.0117023, Final residual = 0.000825005,
No Iterations 6
smoothSolver: Solving for Uy, Initial residual = 0.0308546, Final residual = 0.00302461,
No Iterations 5
smoothSolver: Solving for Uz, Initial residual = 0.0325886, Final residual = 0.0029429,
No Iterations 5
GAMG: Solving for p, Initial residual = 0.0535004, Final residual = 0.0025236, No
Iterations 4
time step continuity errors : sum local = 0.0043995, global = -0.00121763, cumulative = -
0.00293349
smoothSolver: Solving for omega, Initial residual = 0.000334761, Final residual =
1.86261e-05, No Iterations 4
smoothSolver: Solving for k, Initial residual = 0.00264405, Final residual = 0.00014044,
No Iterations 4
bounding k, min: -109.92 max: 382.89 average: 22.3612
ExecutionTime = 177.92 s ClockTime = 180 s

FunctionObject:liftDrag:LDI
Viscous drag: 0.00249183, pressure drag: -0.119703, porous drag: 0
Viscous lift: 2.15173e-06, pressure lift: 0.00669783, porous lift: 0
Total lift: 0.00669998, Front lift: -0.00544654, Rear lift: 0.0121465, Drag: -0.117212
```

Fig.1. 16 Esempio di Output per un solutore SIMPLE

In un solutore SIMPLE per ogni singola iterazione viene mostrato il passo iterativo (in casi stazionari non ha senso parlare di passi temporali), e per ogni variabile risolta, il solutore, i residui iniziali e finali ed il numero di iterazioni interne.

```

Courant Number mean: 0.119372 max: 0.318721
deltaT = 0.00298598
Time = 0.0119159

DILUPPrCG: Solving for Ux, Initial residual = 0.116051, Final residual = 3.29153e-07, No
Iterations 2
DILUPPrCG: Solving for Uy, Initial residual = 0.200717, Final residual = 2.87907e-06, No
Iterations 2
DILUPPrCG: Solving for Uz, Initial residual = 0.281339, Final residual = 2.23115e-06, No
Iterations 2
GAMG: Solving for p, Initial residual = 0.00461653, Final residual = 4.44677e-05, No
Iterations 3
time step continuity errors : sum local = 6.6982e-06, global = -2.24905e-07, cumulative =
-1.13173e-05
GAMG: Solving for p, Initial residual = 0.000194794, Final residual = 5.63733e-07, No
Iterations 5
time step continuity errors : sum local = 8.50057e-08, global = -5.54572e-08, cumulative =
-1.13728e-05
DILUPPrCG: Solving for epsilon, Initial residual = 0.451774, Final residual = 1.05808e-
06, No Iterations 2
DILUPPrCG: Solving for k, Initial residual = 0.418154, Final residual = 2.02963e-06, No
Iterations 2
ExecutionTime = 0.25 s ClockTime = 1 s

```

Fig.1. 17 Esempio di output per un solutore PIMPLE a singola fase

Per un solutore PIMPLE invece ha senso parlare di passo temporale. Vengono mostrati il numero di courant medio e massimo, il time step, e il tempo effettivo della simulazione. Successivamente vengono mostrate, per ogni variabile risolta, il tipo di solutore, i residui finali ed iniziali, ed il numero di iterazioni interne. Come precedentemente spiegato il time step può essere variabile se attivato *adjustTimeStep*.

## 1.7 Parallelizzazione

Dato che può essere utile dividere il costo computazionale tra più processori, si introduce in questo paragrafo il procedimento di parallelizzazione, che sfrutta l'implementazione dei domini pubblici openMPI. Il metodo usato consiste nella decomposizione del dominio, per cui la geometria ed i campi associati vengono divisi e messi nelle cartelle dei processori. Lo scopo è quello di dividere il dominio con il minimo impegno ma in modo tale da garantire una buona soluzione in quanto a costi computazionali.

### 1.7.1 decomposePar

Il file *decomposeParDict* (Fig.1.26), da porre nella cartella *system* del caso, permette di controllare il processo di decomposizione.



```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       decomposeParDict;
}
numberOfSubdomains 4;
method hierarchical;

simpleCoeffs
{
    n            (2 2 1);
    delta        0.001;
}
hierarchicalCoeffs
{
    n            (2 2 1);
    delta        0.001;
    order        vxz;
}
manualCoeffs
{
    dataFile     "decompositionData";
}

distributed false;
roots();

```

Fig.1. 18 Esempio di file *decomposeParDict*

In *numberOfSubdomains* si deve inserire il numero di processori che si vogliono utilizzare durante la simulazione. Il metodo di decomposizione è scelto in *method* tra i seguenti:

1. *simple*, decomposizione geometrica semplice del dominio, il quale viene diviso uniformemente in ogni direzione definita in *simpleCoeffs*
2. *hierarchical*, simile al precedente ma specificando, in *hierarchicalCoeffs*, l'ordine con cui si svolge la suddivisione. E' il metodo suggerito per l'applicazione *snappyHexMesh*.
3. *scotch*, decomposizione completamente automatica, tenta di minimizzare il numero di bordi dei processori senza alcun input geometrico. L'unico input richiesto è *numberOfSubdomains*
4. *ptscotch*, versione parallela di *scotch*, spesso usata quando si avvia *snappyHexMesh* in parallelo, poiché migliora l'esecuzione della soluzione ma può portare a meno robustezza e risultati non voluti
5. *manual*, l'utente deve indicare come distribuire le celle tra i processori attraverso *dataFile* definito in *manualCoeffs*

Attivando il comando *decomposePar* da terminale, si generano all'interno della cartella del caso, le cartelle dei processori (*processor0*, *processor1*, ..., *processorN*, con N numero dei processori scelti in *numberOfSubdomains*), in cui in ognuna si trovano a loro volta una cartella 0 contenente i campi iniziali, una cartella *polyMesh* contenente la griglia e una cartella *constant*.

### **1.7.2 reconstructParMesh**

Per ricostruire la mesh e ricreare il dominio completo per analizzarlo come se fosse uno solo, si usa il comando *reconstructParMesh*, che effettua l'azione opposta a *decomposePar*. La mesh viene ricreata all'interno della cartella *0/polyMesh*. Si noti che questo comando agisce solo sulla mesh e non sui campi.

### **1.7.3 reconstructPar**

Per ricostruire anche i campi invece si usa il comando *reconstructPar* con la quale si uniscono le cartelle temporali da ciascuna cartella *processorN*. Le cartelle temporali verranno create nella cartella principale del caso. Si noti che questo comando agisce solo sui campi e non sulla mesh.

## 1.8 Visualizzazione

Per effettuare le visualizzazioni dei risultati si è utilizzato il software opensource ParaView. Il comando paraFoam da terminale, nella cartella del caso, permette di aprire direttamente i risultati della simulazione in ParaView. In Fig.1.29 si mostra una serie di istruzioni della finestra tipica di ParaView estratto dal manuale di Helyx.

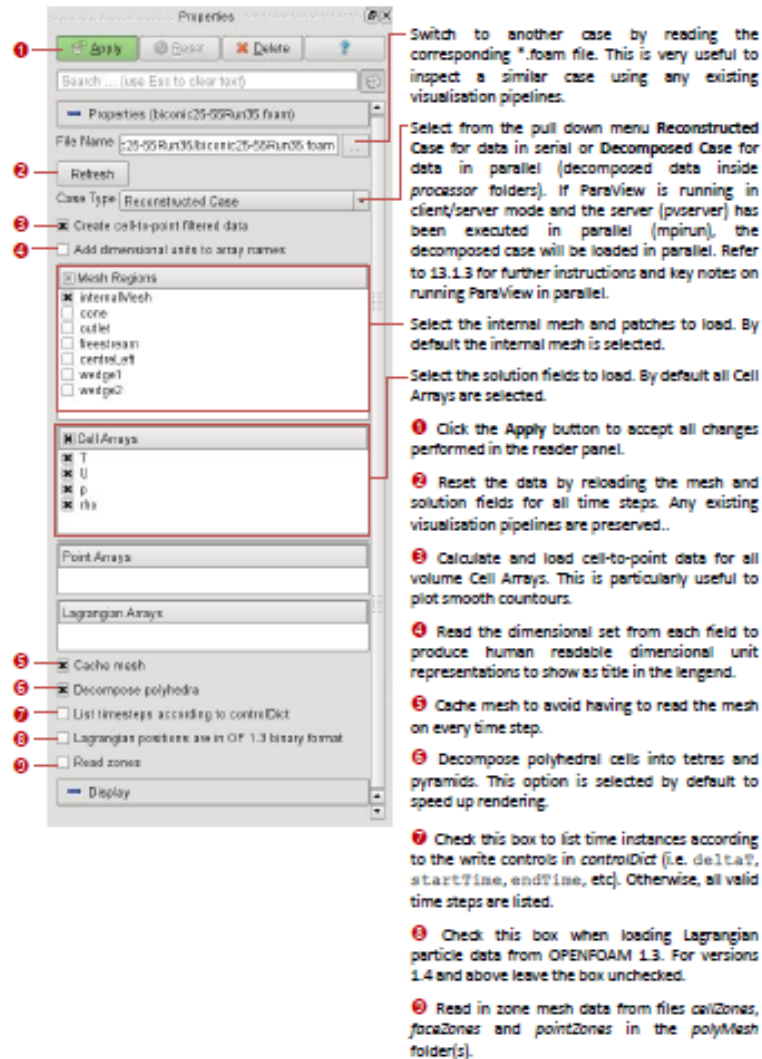


Fig.1. 19 Finestra tipica di ParaView

## 1.9 Pre-Processing avanzato

Tra le funzioni avanzate che Helyx ha implementato ce n'è una di particolare interesse per il nostro caso: *parMapFields*.

### 1.9.1 parMapFields

Questa applicazione è di fondamentale importanza perché permette di mappare soluzioni già ottenute su un caso differente (ha senso solo se una stessa geometria). Per esempio può essere utile in casi in cui si vuole ridurre il tempo di convergenza usando una soluzione intermedia con una mesh meno fitta, per poi mappare i campi ottenuti in una mesh più raffinata. Nel nostro caso quest'applicazione è fondamentale perché si vuole mappare la soluzione del caso stazionario come campo iniziale nel caso non stazionario. *parMapFields* è controllato da *mapFieldsDict* (Fig.1.30) che si trova all'interno della cartella *system* (non c'è di default, va aggiunta manualmente).

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       mapFieldsDict;
}

// List of pairs of target/source patches for mapping
patchMap
{
    patchSource1 patchTarget1
    patchSourceN patchTargetN
};

// List of target patches cutting the source domain
cuttingPatches
{
};

//secondary case
sourceCase     "./path/sourceCaseName";
parallelSource true;
sourceTimeOption latestTime;
sourceTimeValue 1.2;

//primary case
targetTimeOption targetTimeValue;
targetTimeValue 0;

//map properties
consistent     false;
```

Fig.1. 30 Esempio di *mapFieldsDict*

Le sezioni che interessano al nostro caso sono *secondary case* e *primary case*, il resto può essere lasciato di default (si rimanda al manuale di Helyx per maggiori dettagli). In *secondary case* si deve inserire il percorso del caso da cui si vuole prendere la soluzione e scegliere il tempo/iterazione da cui mappare i campi. In *primary case* si inserisce il tempo/iterazione su cui si vuole sovrascrivere la soluzione. *parMapFields* non viene implementato sulla GUI, e perciò va necessariamente eseguito da terminale. Il comando è *parMapFields* da eseguire con *mpirun* se in parallelo.

## Capitolo 2

# Pre-processing

In questo capitolo si illustra il processo di costruzione della mesh. A partire dalla geometria in formato CAD si utilizza l'applicazione *snappyHexMesh* di Helyx per creare sia la mesh di superficie che la mesh di volume. E' necessario trovare quel numero di celle che sia un giusto compromesso tra buona convergenza e esattezza dei risultati, e un tempo di computazione non troppo elevato. L'ordine di grandezza del numero di celle finale da adottare è stato deciso in base ai risultati di lavori precedenti su casi simili e sulla stessa geometria e perciò non è stato ritenuto necessario fare uno studio di validazione della mesh. La mesh di superficie deve essere inoltre ben chiusa per avere dei risultati coerenti.

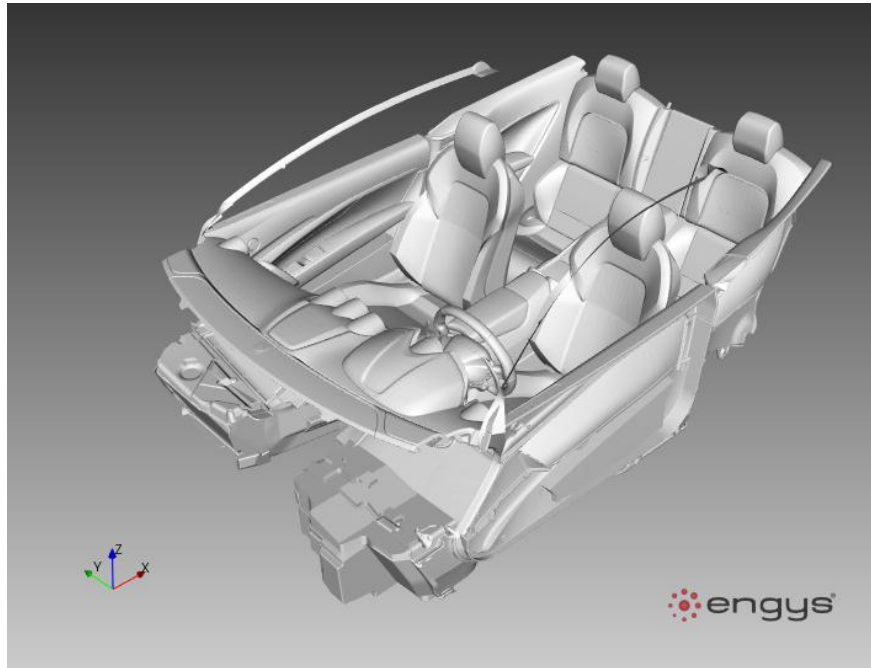
## 2.1 Geometria

I file della geometria che compongono l'abitacolo sono i seguenti (Fig.2.1 - ):

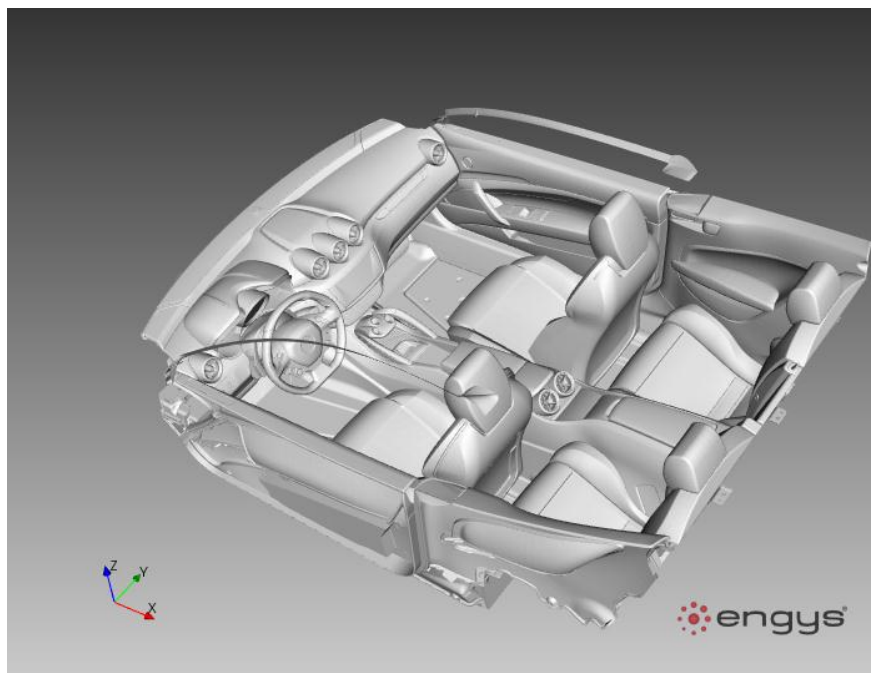
- F151\_bocchette
  - F151\_porte\_plancia\_e\_tappeti
    - F151\_tetto
- F151\_sedili\_e\_sedili laterali
  - F151\_parabrezza
- F151\_lunotto
  - F151\_finestrini\_lato\_guidatore
  - F151\_finestrini\_lato\_passeggero
- F151\_integrazioni
- F151\_inlet\_centrali
- F151\_inlet\_destro
- F151\_inlet\_sinistro
- F151\_inlet\_posteriori

- F151\_outlet\_destro
  - F151\_outlet\_sinistro
  - F151\_tunnel

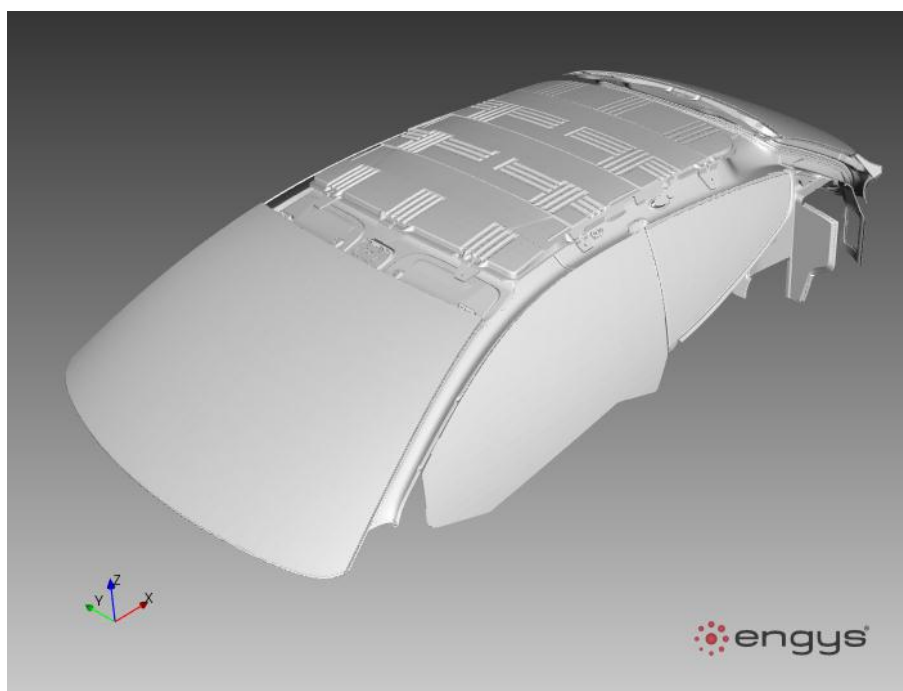
A queste vanno aggiunte poi le patch dei manichini, creati tramite l'apposito ambiente di CATIA



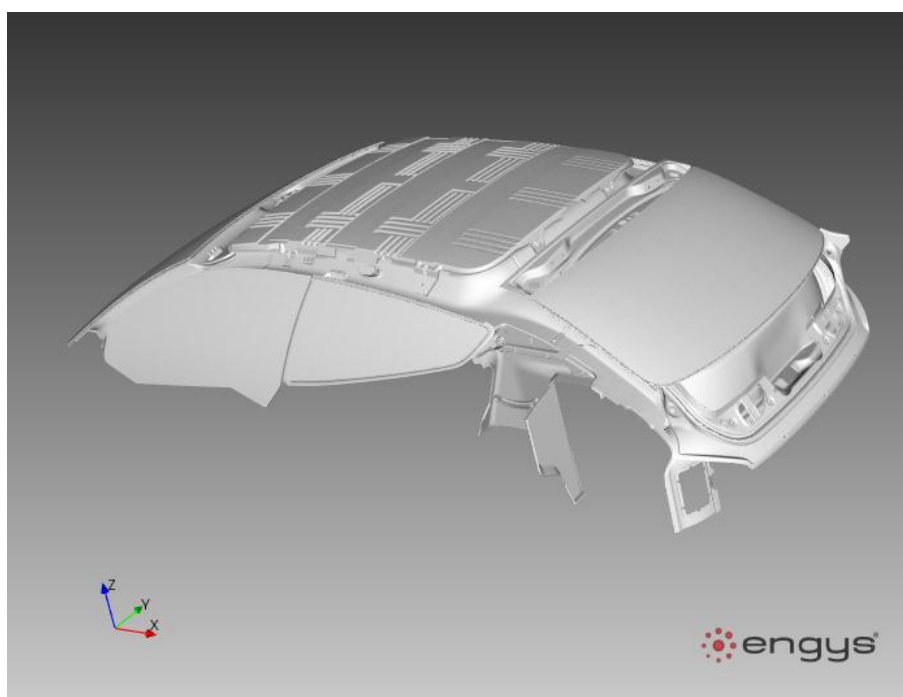
*Fig.2. 20 Porte, Plancia. Tappeti, Tunnel, Sedili e Bocchette*



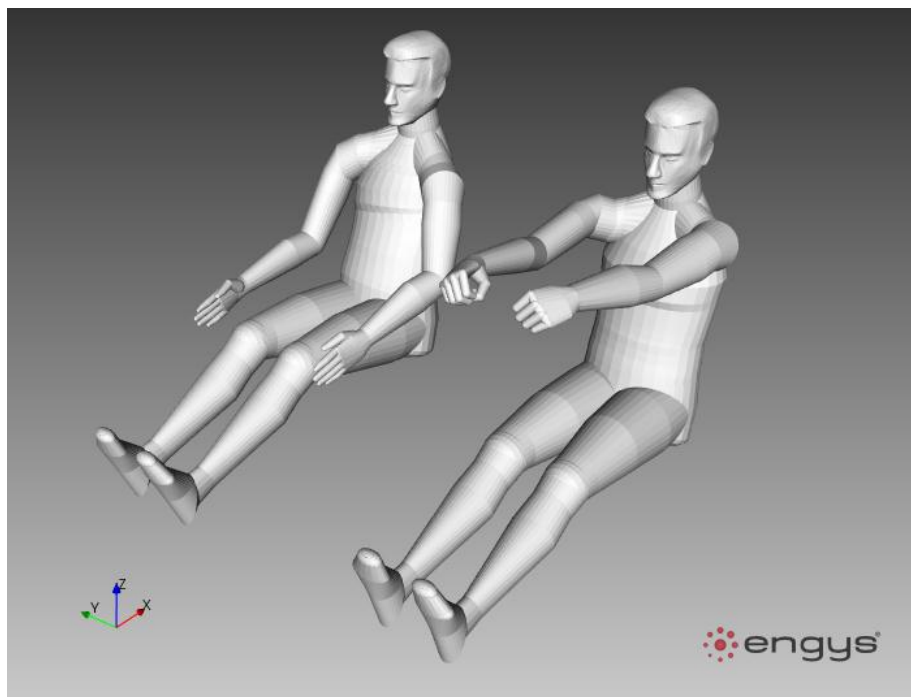
*Fig.2. 21 Porte, Plancia. Tappeti, Tunnel, Sedili e Bocchette (altra prospettiva)*



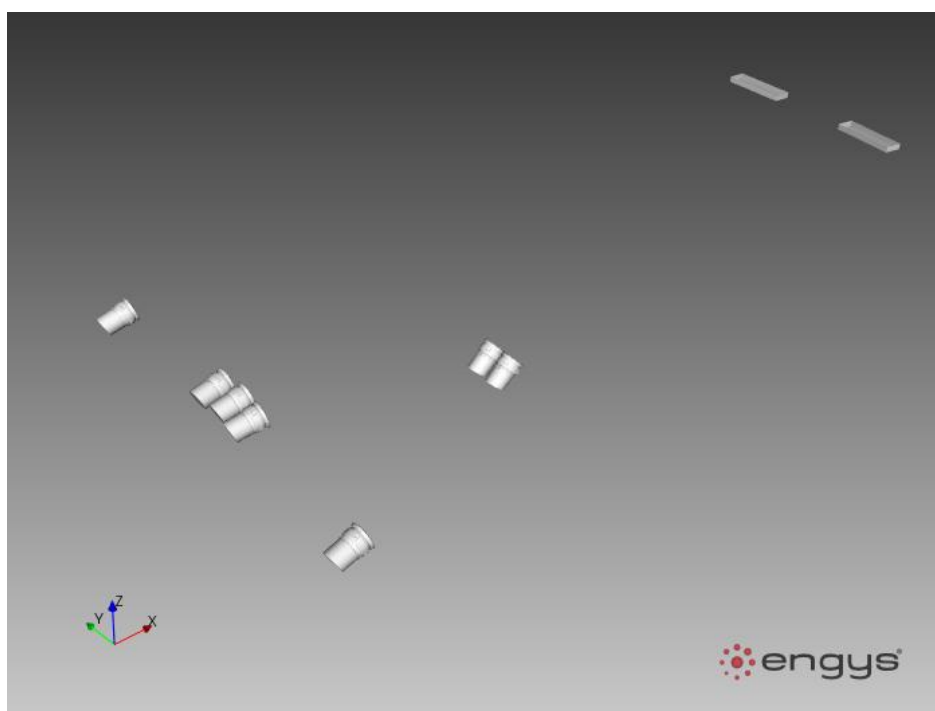
*Fig.2. 22 Parabrezza, Finestrini, Lunotto e Tetto*



*Fig.2. 23 Parabrezza, Finestrini, Lunotto e Tetto (altra prospettiva)*



*Fig.2. 24 Manichini del Guidatore e del Passeggero*



*Fig.2. 25 Bocchette con Inlet, ed i box Outlet*

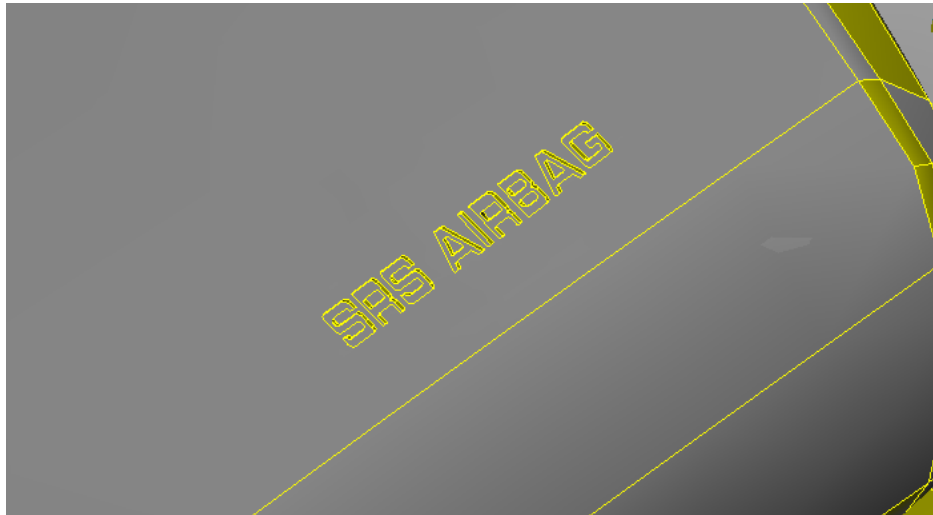


L'applicazione *snappyHexMesh* di Helyx, utilizzata per creare la mesh, dispone di alcuni strumenti, come crack detection e wrapping, che permettono all'utente di semplificare il processo di discretizzazione, automatizzando alcune operazioni da fare altrimenti a mano. Il primo permette di chiudere i fori presenti nella geometria che rientrano sotto certe tolleranze geometriche, mentre il secondo permette, oltre a tappare i buchi come il primo, ma con criteri diversi dal crack detection, di tagliare quelle parti che non partecipano come contorno del dominio contenente la mesh di volume (essendo questo un caso di fluidodinamica interna, ovvero un abitacolo), e che quindi non utili alla simulazione. Questi strumenti però non sono affatto perfetti, e perciò è stato comunque necessario intervenire manualmente sulla geometria. I file della geometria sono stati forniti in formato .igs, e sono stati modificati con i software di ANSA e CATIA.

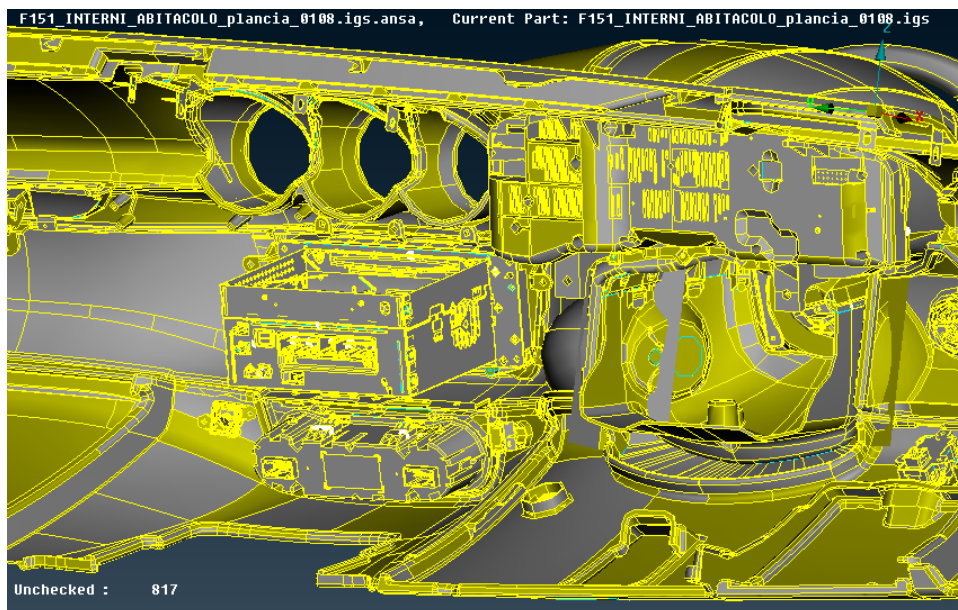
Sono state fatte le seguenti operazioni:

- Sono stati tolti dalla geometria alcuni dettagli presenti nelle superfici che si affacciavano al dominio, non incisivi ai fini del risultato, ma che complicavano la mesh (per esempio le scritte sul volante e sul cruscotto e alcuni pulsanti minori, Fig.2.7)
- Sono state pulite le superfici che si affacciavano alla parte esteriore dell'abitacolo e quindi non importanti per un caso di fluidodinamica interna (Fig.2.8). Il wrapping fa il suo dovere ma non è abbastanza accurato.
- La geometria fornita presentava delle mancanze: vi erano alcuni spazi tra le patch, troppo grandi per essere chiuse con wrapping o ancora a maggior ragione dal crack detection. Sono state perciò costruite delle superfici ausiliari che chiudessero questi spazi. L'insieme di queste superfici sono state implementate nella geometria finale come una patch ulteriore chiamata *integrazioni* (Fig.2.).

Fatto ciò si è convertito i file da formato CAD a formato stereolitografico (.stl), l'unico che viene riconosciuto da Helyx.



*Fig.2. 26 Esempio di dettagli da pulire sulle superfici rivolte verso il dominio*



*Fig.2. 27 Plancia prima della pulizia*

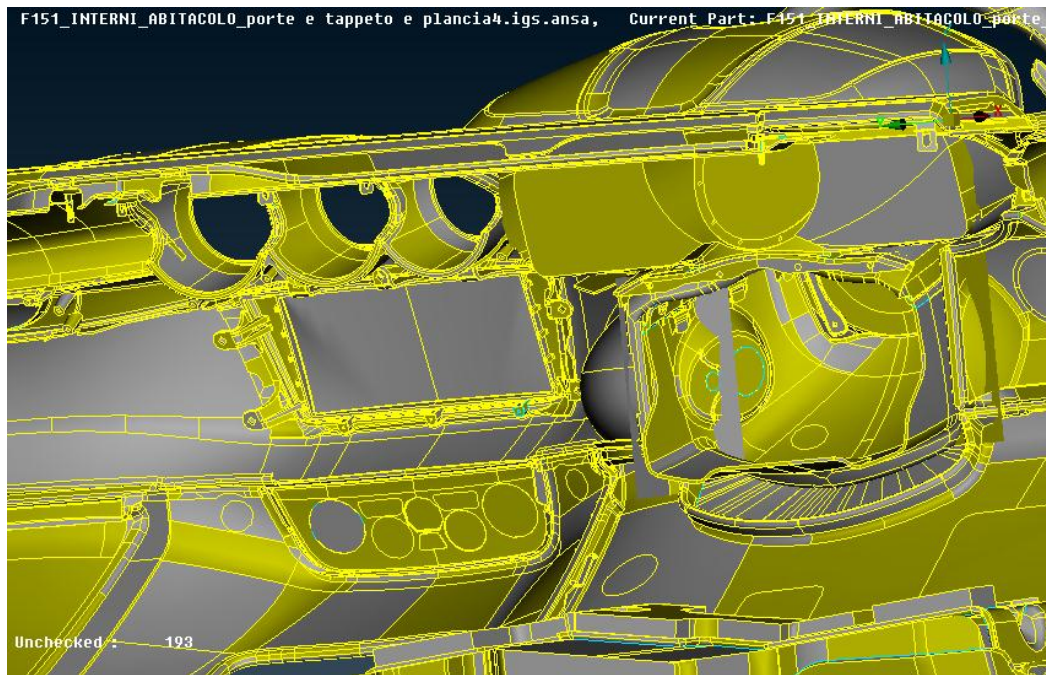


Fig.2. 28 Plancia dopo la pulizia

## 2.2 Costruzione della mesh

Si procede quindi a impostare *snappyHexMesh* per costruire la mesh. L'applicazione è stata descritta nel capitolo 1, si usa quindi la GUI di Helyx per semplificare il processo di impostazione, che altrimenti sarebbe dovuta esser fatta a mano tramite le *dict*.

La schermata iniziale di Helyx è visibile in Fig.2.10. Si scelgono il numero di processori, la loro gerarchia e la cartella in cui verrà creato il caso.

Cliccando sulla casella STL (Fig.2.11), si importa la geometria \*.stl, che verrà posizionata nella cartella *constant/triSurface*. Durante questa operazione è possibile scalare la geometria nelle 3 dimensioni. E' necessario scalare le dimensioni ogni qualvolta i file \*.stl perdono l'informazione delle grandezza (ciò che erano millimetri li legge come metri). In questo caso si è dovuto fare una riduzione di 1000 volte in ogni dimensione (Fig.2.12).

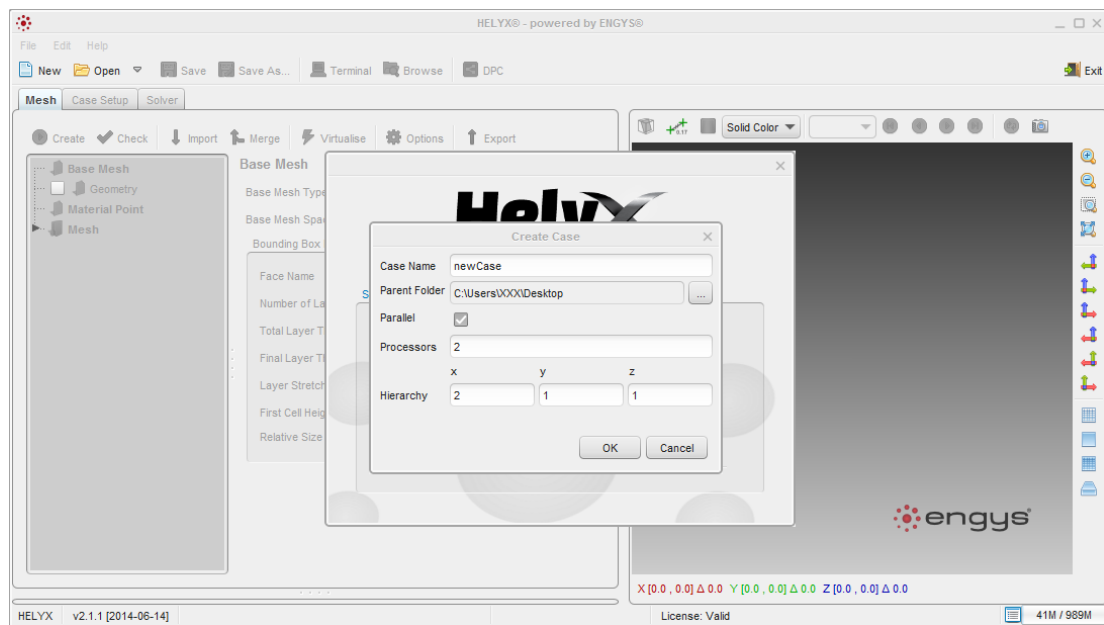


Fig.2. 29 Schermata iniziale di Helyx

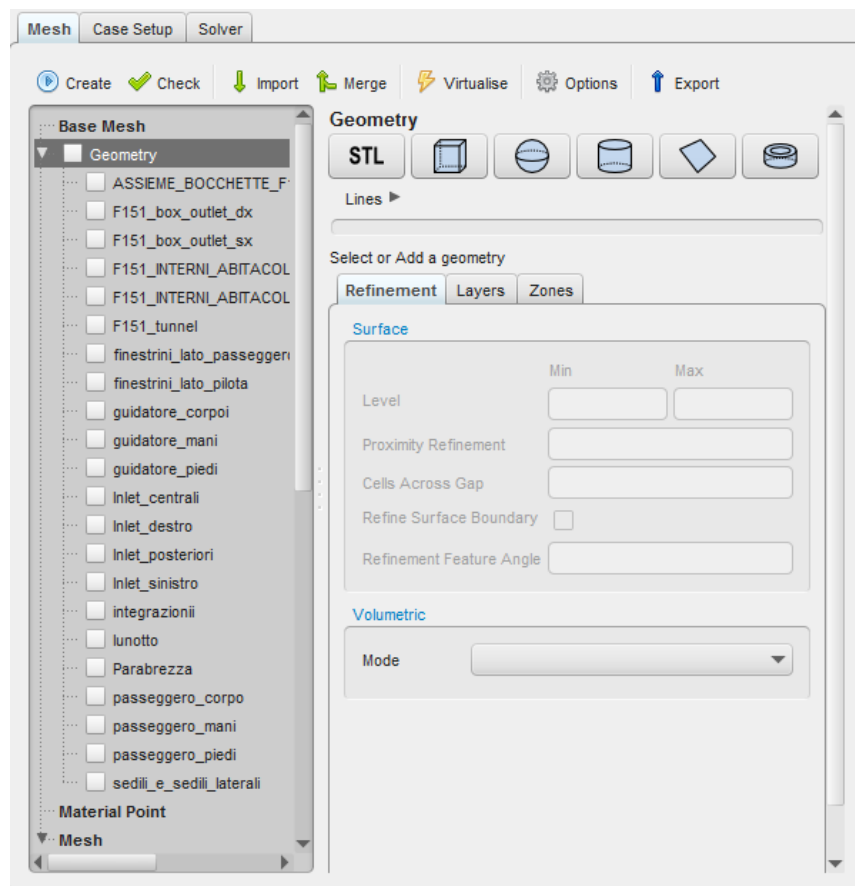


Fig.2. 30 Importazione della geometria \*.stl

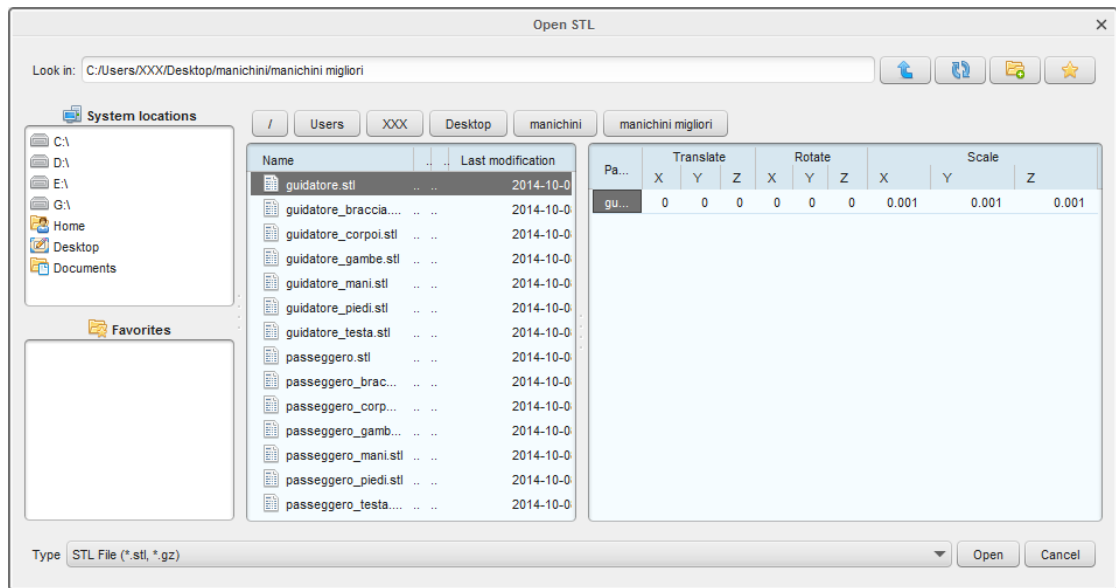


Fig.2. 31 Scalatura della geometria durante l'importazione dei file \*.stl

La schermata successiva (Fig.2.13) è quella relativa all'ambiente mesh, in cui saranno impostati i parametri di generazione della griglia.

Si scelgono il tipo di mesh base, in *Base Mesh Type*, e la sua spaziatura, in *Base Mesh Spacing* (se è pari a 0.1, verrà creata una mesh di base con spaziatura 0.1 m). Questi sono i requisiti necessari per l'applicazione *blockMesh*.

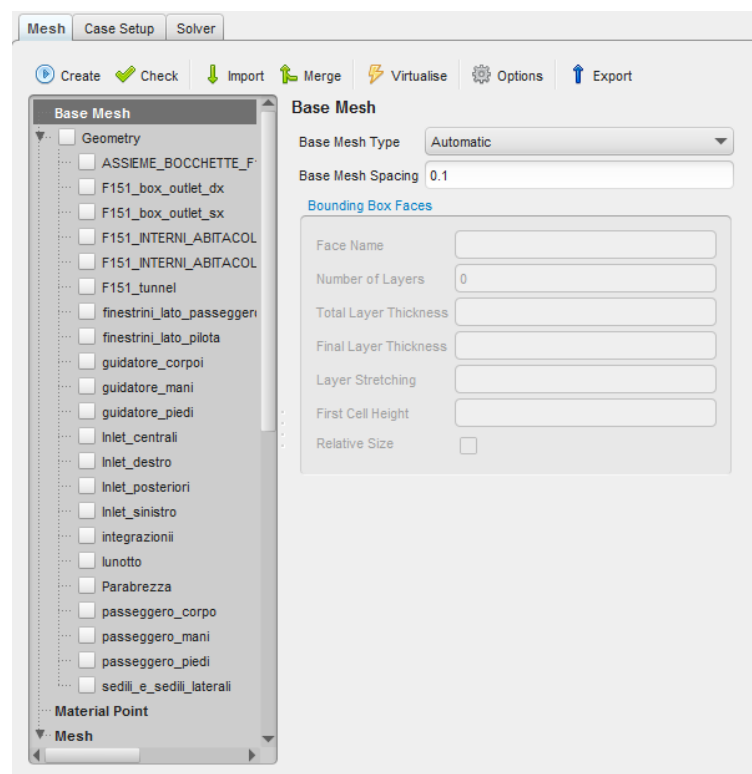


Fig.2. 32 Ambiente di generazione della griglia

Selezionando tutta la geometria è possibile fare sia i raffinamenti superficiali, che quelli volumetrici. Per impostare i raffinamenti superficiali si devono selezionare i livelli massimo e minimo di raffinamento (ad esempio come nel nostro caso, dato un block mesh di 0.1, se il livello minimo è 3 e quello massimo è 4, vorrà dire che la mesh avrà risoluzione minima pari a 0.0125 m e risoluzione massima, ovvero dove la curvatura supera un determinato valore, pari a 0.00625 m. Questo è in accordo con la Fig.1.9)

I *layers* non sono stati considerati per due motivi:

- Gli strati limite non sono il fenomeno fisico che si vuole andare ad indagare
- La generazione di prismi nella griglia, al fine di caratterizzare lo strato limite, comporta costi computazionali onerosi

E' possibile ovviamente andare a selezionare una specifica patch per impostare i raffinamenti superficiali e il *Refinement Feature Angle* (Fig.2.14), che è proprio quello che è stato fatto. Nella tabella 2.1 si riportano i raffinamenti superficiali, minimo e massimo, e il *Refinement Feature Angle* per ogni patch.

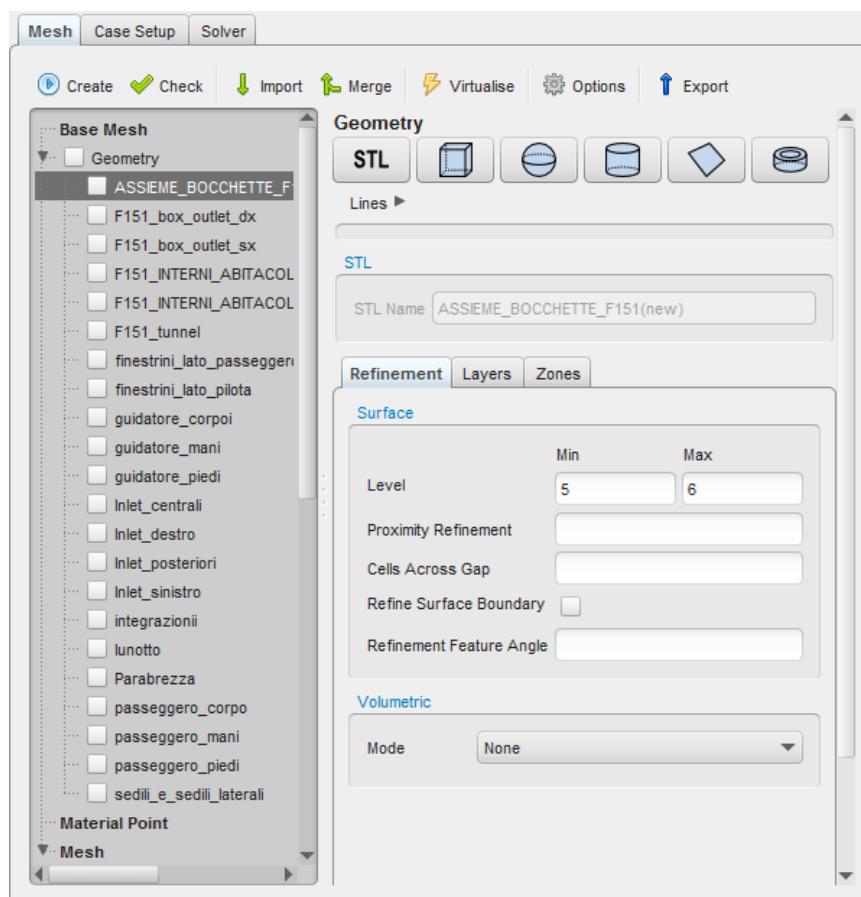


Fig.2. 33 Raffinamenti superficiali e volumetrici

Patch	Liv Min	Liv Max	Refinement Feature Angle
F151_bocchette	5	6	10
F151_porte_plancia_e_tappeti	3	4	10
F151_tetto	3	4	10
F151_sedili_e_sedili_laterali	3	4	10
F151_parabrezza	3	4	10
F151_lunotto	3	4	10
F151_finestrini_lato_guidatore	3	4	10
F151_finestrini_lato_passeggero	3	4	10
F151_integrazioni	3	4	10
F151_inlet_centrali	3	4	10
F151_inlet_destro	3	4	10
F151_inlet_sinistro	3	4	10
F151_inlet_posteriori	3	4	10
F151_outlet_destro	3	3	10
F151_outlet_sinistro	3	3	10
F151_tunnel	3	4	10
F151_guidatore_corpo	3	4	10
F151_guidatore_man	3	4	10
F151_guidatore_piedi	3	4	10
F151_passeggero_corpo	3	4	10
F151_passeggero_man	3	4	10
F151_passeggero_piedi	3	4	10

Tab.1. 1 Livello di raffinamento superficiale delle patch

Nelle opzioni avanzate si può agire su diversi aspetti, come ad esempio attivare il *crack detection* e il *wrapping* (Fig.2.15 e Fig.2.16). Per le opzioni di wrapping si sceglie una *Cut Off Pressure* di 200, una deviazione standard *Sigma* di 3, e un *Volume Distance* di 1.

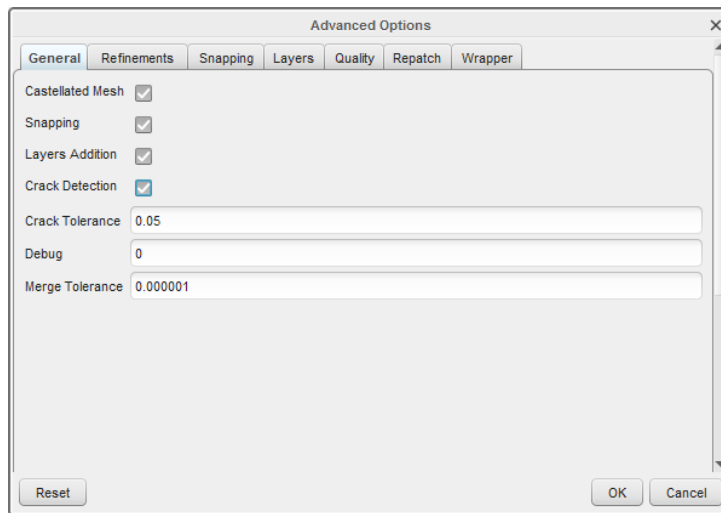


Fig.2. 34 Opzioni avanzate

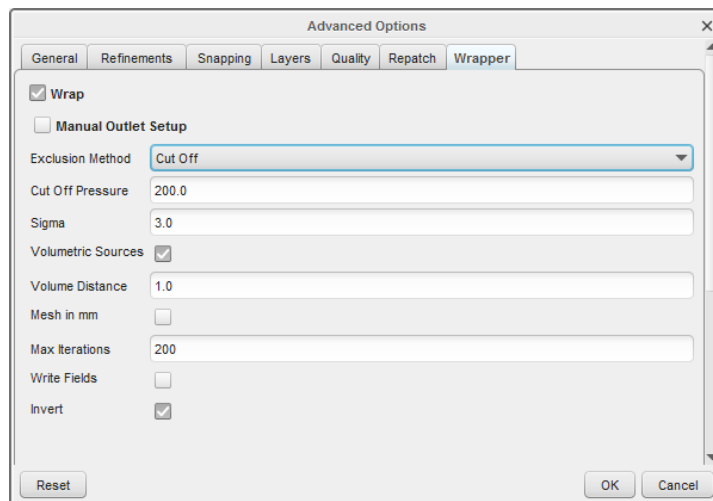


Fig.2. 35 Opzioni di Wrapping

Per definire la zona che dovrà essere discretizzata, si mette il *material point* all'interno del volume dell'abitacolo (Fig.2.17). Infine, cliccando su *Create*, si avvia la generazione della griglia.



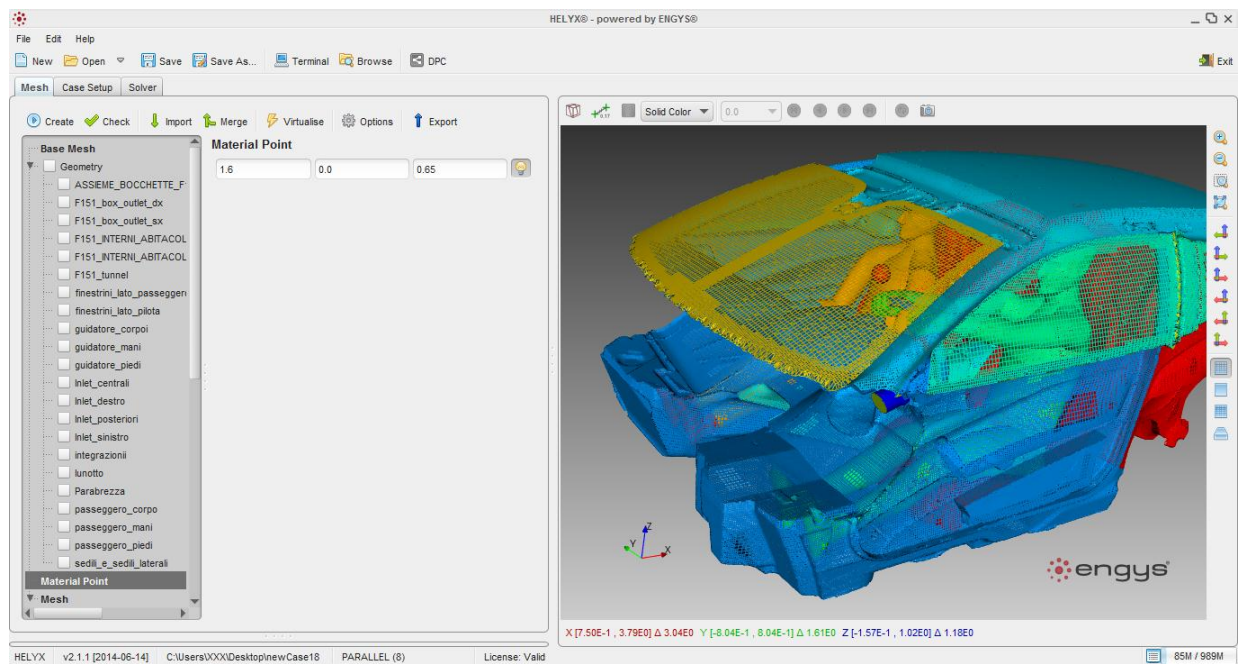


Fig.2. 36 Posizionamento del Material Point

Una volta creata la mesh è possibile vederne le caratteristiche (Fig.2.18) ed, eventualmente, effettuare i controlli con checkMesh e snappyCheckMesh.

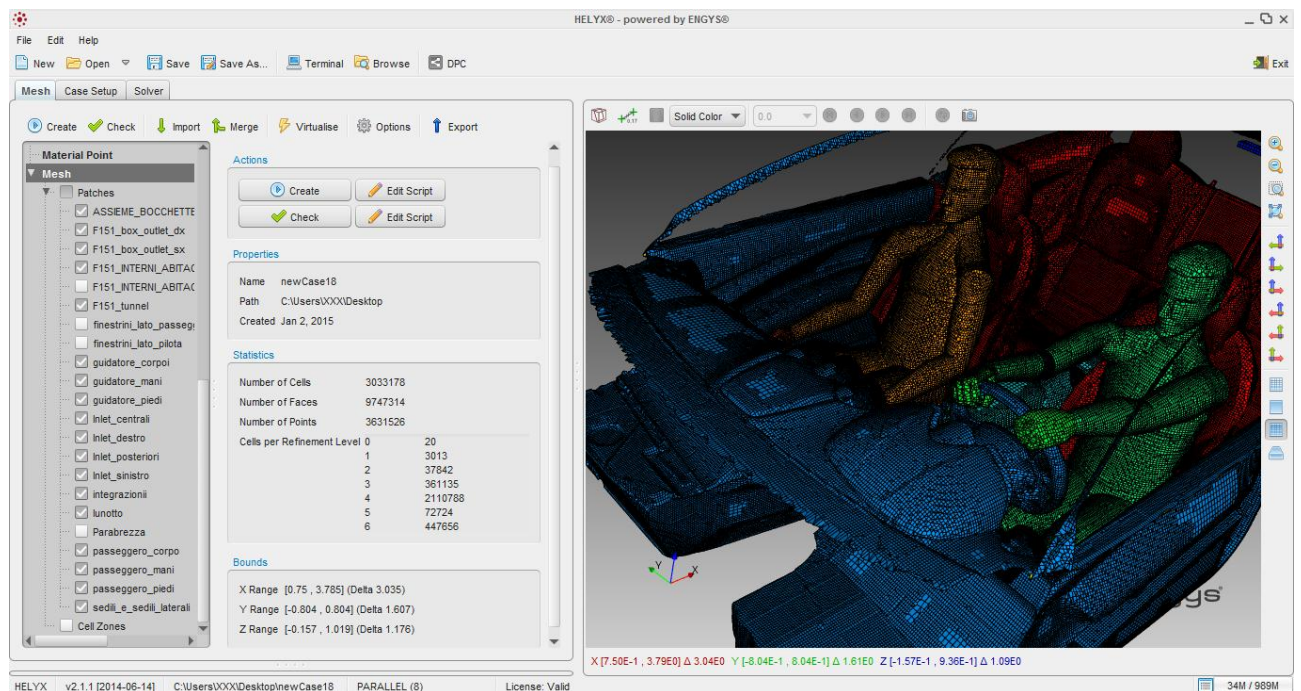


Fig.2. 37 Visualizzazione della mesh e le sue caratteristiche.

La mesh finale, utilizzata per la simulazione sia stazionaria che non stazionaria, risulta con le seguenti caratteristiche:

- Una mesh di volume di 6469735 celle
- Una mesh di superficie di 742179 celle

Per un totale di 7211914 celle.

## Capitolo 3

# Solving

Una volta costruita la mesh si procede allo studio del modello fisico e quindi all'impostazione del solutore. L'obiettivo di questo lavoro è studiare il transitorio del flusso d'aria all'interno dell'abitacolo in condizioni estreme di caldo. Perciò la prima cosa da fare è impostare una condizione di partenza per il transitorio a bocchette chiuse (senza aria condizionata). Per fare ciò si procede ad impostare un caso stazionario (a regime) dell'abitacolo con carico solare e temperature esterne elevate (Dubai, 15 agosto, 12 a.m - Temperatura esterna 40°). Successivamente si utilizza il risultato della soluzione stazionaria come condizione iniziale per far partire il transitorio, stavolta con le bocchette aperte e aria condizionata accesa. Il caso è un problema di fluidodinamica non stazionario (per il primo caso stazionario) incomprimibile, con modello radiativo e con l'implementazione di un carico solare. Si usa perciò un modello computazionale che utilizzi le equazioni RANS (Reynolds Averaged Navier Stokes) e con il modello di turbolenza k-epsilon realizable. L'applicazione di Helyx adatta al caso è il *buoyantBoussinesqSimpleFoam* per il caso stazionario e il *buoyantBoussinesqPimpleFoam* per il caso non stazionario e sono stati descritti nel Capitolo 1.

### 3.1 Modello radiativo e solare di Helyx

Il modello radiativo e il modello solare di Helyx presentano alcune lacune, le quali non hanno permesso una riproduzione fedele della situazione con un caso reale. La radiazione in Helyx è trattata usando il metodo ai volumi finiti DOM (Discrete Ordinate Method). In questo modello, l'equazione di trasferimento della radiazione (RTE, equazione 3.1) per un mezzo emissivo, assorbente, con scattering, nella posizione  $\vec{r}$  e in direzione  $\vec{s}$  è risolta usando un numero finito di angoli solidi, ciascuno associato con un vettore direzione fissato nel sistema cartesiano globale.

$$\frac{dI(\vec{r}, \vec{s})}{ds} = (\alpha + \sigma_s)I(\vec{r}, \vec{s}) = \alpha n^2 \frac{\sigma T^2}{\pi} + \frac{\sigma_s}{4\pi} \int_0^{4\pi} I(\vec{r}, \vec{s}') \Phi(\vec{r}, \vec{s}, \vec{s}') d\Omega' \quad (\text{Eq. 3.1})$$

dove:

$I$ =intensità totale della radiazione,  $\text{W}/\text{m}^2$

$\vec{r}$ =vettore posizione

$\vec{s}$ =vettore direzione

$\alpha$ =coefficiente di assorbimento lineare,  $\text{m}^{-1}$

$\sigma_s$ =coefficiente di scattering,  $\text{m}^{-1}$

$n$ =indice di rifrazione

$T$ =temperatura locale, K

$\sigma$ =costante di Boltzmann,  $5.67 \cdot 10^{-8} \text{W}/\text{m}^2 \text{K}^4$

$\Phi$ =funzione di fase di scattering

$\vec{s}'$ =vettore direzione dello scattering

$\Omega'$ =angolo solido

Nel modello implementato da Engys, lo scattering (processo fisico per cui alcune forme di radiazione, come la luce, vengono forzate a deviare la propria traiettoria rettilinea a cause del loro passaggio in mezzi che presentano locali disuniformità) e l'assorbimento dovuto a fluidi circostanti sono ignorati. Nella sotto-sezione *constant* del file *caseSetupDict* (Fig.3.1), è possibile modificare due parametri del modello: *nPhi*, in cui specifica il numero di angoli solidi nella direzione azimutale (di default questo valore è pari a 2) e *nTheta*, in cui si specifica il numero di angoli solidi nella direzione polare (di default questo valore è pari a 4). Più alti saranno i valori di questi due parametri e maggiore sarà l'accuratezza, con lo svantaggio, però di aumentare i costi computazionali. Si può anche impostare la frequenza di risoluzione della RTE nell'opzione *solverFreq*: l'equazione della radiazione verrà risolta dopo ogni intervallo di passi temporali/iterazioni specificati in questa opzione. Non è possibile modificare altro se non questi tre parametri.

```
constant
{
    radiationProperties
    {
        fvDOMCoeffs
        {
            nPhi      3;
            nTheta    5;
        }
        solverFreq    50;
    }
}
```

Fig.3. 1 Proprietà radiative nella sotto-sezione *constant* del file *caseSetupDict*

Il modello solare di Helyx è gestito da una funzione dedicata (Fig.3.2).

```
functions
{
    solarRadiation
    {
        type solarRadiation;
        solarIntensity 500;
        solarDirection ( 1 1 0 );
        transmissivity (wall window 0.8);
    }
}
```

Fig.3. 2 Esempio di funzione solarRadiation

S'impostano l'intensità del carico solare,  $\text{W/m}^2$ , la direzione del calcolo solare, specificando il vettore che punta da sole verso il dominio di calcolo, e la transmissività solare di tutte le patch semi-trasparenti o trasparenti, ovvero quanto carico solare passa attraverso le patch. Ad esempio il carico solare che passerà attraverso la finestra (wall\_window) di Fig.3.2 sarà pari a  $0.8 \times 500 = 400 \text{ W/m}^2$ .

La semplicità dei modelli radiativo e solare di Helyx, ha come semplice conseguenza, la carenza di accuratezza dei risultati e quindi questi subiscono varie approssimazioni. Per la precisione nel modello radiativo non è possibile simulare un modello a bande e, quindi, anche l'effetto serra, che è una delle caratteristiche termiche fondamentali di un abitacolo di un automobile. Inoltre nel modello solare, cambiando la direzione dei raggi solari, cambiano solo le zone che esse colpiscono ma non cambia l'intensità del carico solare che le raggiunge. Tuttavia nella realtà l'intensità cambia perché le proprietà radiative del vetro variano con l'angolo di incidenza dei raggi solari.

## 3.2 Caso Stazionario

Il primo passo del nostro caso è stato quello di trovare il campo di temperature all'interno dell'abitacolo in condizioni di regime se sottoposto ad un determinato carico solare (Dubai, 15 agosto, mezzogiorno, temperatura ext  $40^\circ$ ) con aria condizionata spenta. Il carico solare in queste determinate condizioni è stato determinato manualmente dalle normative di ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers) del 2009.

### 3.2.1 ASHRAE 2009

In questo modello la costante solare  $E_{sc}$  è definita come l'intensità della radiazione solare su una superficie normale ai raggi del sole, appena sopra l'atmosfera della Terra, alla distanza media Terra-Sole. Un valore di uso frequente è quello che è stato proposto dalla World Meteorological Organization nel 1981:

$$E_{sc} = 1397 \text{ W/m}^2 \quad (\text{Eq.3.2})$$

Poiché l'orbita terrestre è leggermente ellittica, il flusso radiante extraterrestre,  $E_0$ , varia durante l'anno, raggiungendo un massimo di  $1412 \text{ W/m}^2$  all'inizio di gennaio, quando la Terra è più vicina al Sole e un minimo di  $1322 \text{ W/m}^2$ , all'inizio di luglio, quando la Terra è più lontana dal Sole. La radiazione solare extraterrestre su una superficie normale ai raggi solari può essere approssimata con la seguente equazione:

$$E_0 = E_{sc} \left\{ 1 + 0.033 \cos \left[ \frac{360^\circ(n-3)}{365} \right] \right\} \quad (\text{Eq.3.3})$$

dove  $n$  è il numero del giorno (1= primo gennaio, 32 = primo febbraio).

La velocità orbitale della Terra varia durante tutto l'anno, quindi l'ora solare apparente (AST) varia un po' dal tempo medio ottenuto da un orologio. Questa variazione è detta equazione del tempo (ET), che si esprime in minuti, ed è approssimata dalla seguente formula:

$$ET = 2.2918[0.0075 + 0.1868 \cos(\Gamma) - 3.2077 \sin(\Gamma) - 1.4615 \cos(2\Gamma) - 4.089 \sin(2\Gamma)] \quad (\text{Eq.3.4})$$

dove:

$$\Gamma = 360^\circ \frac{n-1}{365}$$

La conversione tra il tempo locale e il tempo solare prevede due fasi: l'equazione del tempo si aggiunge all'ora locale e poi si aggiunge una correzione sulla longitudine. Questa correzione è di quattro minuti di tempo per ogni grado di differenza tra la longitudine locale (LON) e la longitudine del meridiano locale standard (LSM) per quel fuso orario. Quindi AST è legato a LST come segue:

$$AST = LST + \frac{ET}{60} + (LON - LSM)/15 \quad (\text{Eq.3.5})$$

dove:

AST è il tempo solare apparente in decimi d'ora

LST è il tempo locale in decimi d'ora

LON è la longitudine locale, °E da Greenwich (negativo se ad ovest di Greenwich)

LSM è la longitudine del meridiano locale standard per quel fuso orario, °E da Greenwich.

Poiché il piano equatoriale della Terra è inclinato di un angolo di 23.45° rispetto al piano orbitale, la declinazione solare  $\delta$  (angolo tra la linea Terra-Sole e il piano equatoriale) varia durante tutto l'anno. Tale variazione provoca il cambiamento delle stagioni. La seguente equazione fornisce  $\delta$  con sufficiente precisione:

$$\delta = 23.45 \sin 360^\circ \left( \frac{n+284}{365} \right) \quad (\text{Eq.3.6})$$

In Fig.3.3 sono riportati i dati del giorno 21 di ogni mese.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Day of year	21	52	80	111	141	172	202	233	264	294	325	355
$E_o$ , W/m <sup>2</sup>	1410	1397	1378	1354	1334	1323	1324	1336	1357	1380	1400	1411
Equation of time (ET), min	-10.6	-14.0	-7.9	1.2	3.7	-1.3	-6.4	-3.6	6.9	15.5	13.8	2.2
Declination $\delta$ , degrees	-20.1	-11.2	-0.4	11.6	20.1	23.4	20.4	11.8	-0.2	-11.8	-20.4	-23.4

Fig.3. 3 Dati astronomici approssimati per il giorno 21 di ogni mese

La posizione del sole nel cielo è espressa in termini di altitudine solare sopra l'orizzonte e l'azimut solare misurato da sud (Fig.3.4). L'angolo di altitudine solare  $\beta$  è definito come l'angolo tra il piano orizzontale e la linea che congiunge il Sole al centro della Terra. Il suo valore varia da 0°, quando il sole è sull'orizzonte, a 90° quando il sole è allo zenit. Valori negativi corrispondono alla notte. L'angolo di azimut solare  $\phi$  è definito come lo spostamento angolare da sud della proiezione, sul piano orizzontale, della linea Terra-Sole. Per convenzione, si conta positivo per ore pomeridiane e negativo per ore del mattino. Gli angoli di altitudine e azimut solari dipendono dalla latitudine locale  $L$  (°N, negativa nell'emisfero meridionale), dalla declinazione solare  $\delta$ , che è una funzione della data, e dall'angolo orario  $H$ , definito come lo spostamento angolare del Sole ad est o ad ovest del meridiano locale, dovuto alla rotazione della Terra, ed espresso in gradi:

$$H = 15(AST - 12) \quad (\text{Eq.3.7})$$

H è zero a mezzogiorno, positivo nel pomeriggio e negativo al mattino. L'equazione seguente permette di determinare  $\beta$  conoscendo L,  $\delta$  e H:

$$\sin \beta = \cos L \cos \delta \cos H + \sin L \sin \delta \quad (\text{Eq.3.8})$$

Quando  $H=0$  (mezzogiorno), il Sole raggiunge la massima altezza:

$$\beta_{max} = 90^\circ - |L - \delta| \quad (\text{Eq.3.9})$$

L'angolo di azimuth è determinato da suo seno e dal suo coseno:

$$\sin \phi = \sin H \cos \delta / \cos \beta \quad (\text{Eq.3.10})$$

$$\cos \phi = (\cos H \cos \delta \sin L - \sin \delta \cos L) / \cos \beta \quad (\text{Eq.3.11})$$

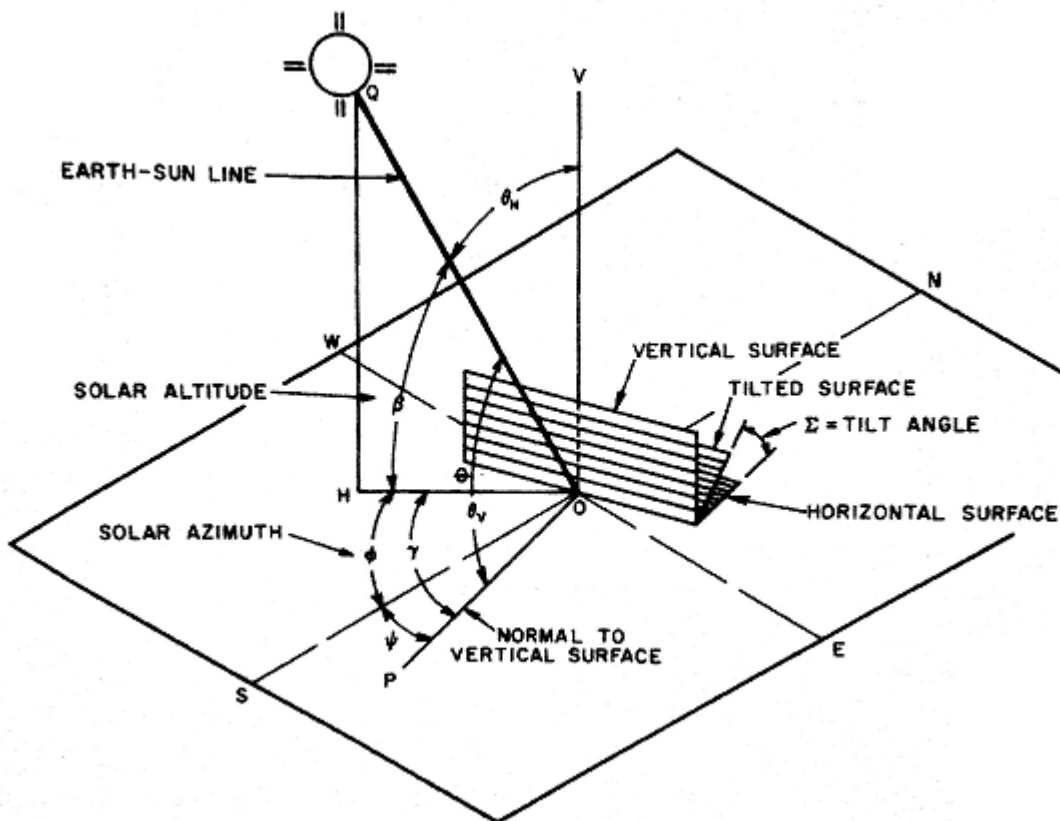


Fig.3. 4 Angoli solari per superfici verticali e orizzontali



La radiazione solare in una giornata limpida è definita dalle sue componenti diretta e diffusa. La componente diretta rappresenta la parte di radiazione solare proveniente direttamente dal disco solare, mentre la componente diffusa rappresenta la radiazione proveniente dal resto del cielo. Queste due componenti sono calcolate nel seguente modo:

$$E_b = E_0 \exp(-\tau_b m^{ab}) \quad \text{Eq.3.12}$$

$$E_d = E_0 \exp(-\tau_d m^{ad}) \quad \text{Eq.3.13}$$

dove:

$E_b$ =radiazione solare normale diretta, misurata perpendicolarmente ai raggi solari

$E_d$ =radiazione solare orizzontale diffusa, misurata su un piano orizzontale

$m = 1/\sin\beta + 0.50572(6.07995 + \beta) - 1.6364$  = rapporto tra la massa di atmosfera nel percorso Terra-Sole e la massa che si avrebbe se il sole fosse allo zenit

$\tau_b$ =profondità ottica diretta

$\tau_d$ =profondità ottica diffusa

$ab$  e  $ad$  = esponenti diretto e diffuso dalla massa d'aria

I valori di  $\tau_b$  e di  $\tau_d$ , che variano durante l'anno e con la posizione rappresentano la dipendenza della radiazione solare con cielo limpido alle condizioni locali, come altitudine, aerosol e contenuto di acqua precipitabile.  $\tau_b$  e  $\tau_d$  sono tabulati per moltissime località e per ogni giorno 21 di ogni mese. I valori di  $ab$  e  $ad$  sono legati alle profondità ottiche nel seguente modo:

$$ab = 1.219 - 0.043\tau_b - 0.151\tau_d - 0.204\tau_b\tau_d \quad (\text{Eq.3.14})$$

$$ad = 0.202 + 0.852\tau_b - 0.007\tau_d - 0.357\tau_b\tau_d \quad (\text{Eq.3.15})$$

Le equazioni 3.12 – 3.15 descrivono una semplice parametrizzazione di un sofisticato modello di radiazione e forniscono previsioni accurate di  $E_b$  e  $E_d$ , anche in luoghi dove l'atmosfera è molto nebbiosa e umida per la maggior parte del tempo.

Le equazioni precedenti sono utili solo per calcolare la radiazione solare normale diretta e quella solare diffusa, ma se le superfici sono di orientamento arbitrario le cose cambiano. La superficie può essere verticale (ad esempio, pareti e finestre) o inclinata (ad esempio, lucernari, dispositivi solari attivi o finestrini di automobili). L'orientamento di una superficie ricevente è caratterizzato dal suo angolo d'inclinazione e dal suo azimut, come mostrato in Fig.3.5. L'angolo d'inclinazione  $\Sigma$  è l'angolo tra la superficie e il piano orizzontale. Il suo valore è compreso tra  $0^\circ$  e  $180^\circ$  e, molto spesso, è tra  $0^\circ$  (orizzontale) e  $90^\circ$  (verticale). L'angolo di azimut superficiale  $\psi$  è definito come lo spostamento da sud della proiezione, sul piano orizzontale, della normale alla superficie. Le superfici che si affacciano ad ovest hanno un azimut di superficie positivo, quelle che si affacciano ad est hanno un valore negativo.

Orientation	N	NE	E	SE	S	SW	W	NW
Surface azimuth $\psi$	$180^\circ$	$-135^\circ$	$-90^\circ$	$-45^\circ$	0	$45^\circ$	$90^\circ$	$135^\circ$

Fig.3. 5 Orientazioni della superficie e corrispondente angolo di azimut superficiale, misurato da sud

Gli angoli di azimut di superficie per orientamenti comuni sono riassunti in Fig.3.5. Si noti che l'angolo di azimut superficiale è definito come relativo a sud in entrambi li emisferi boreale e australe (altri calcolatori usano una diversa impostazione e, quindi, bisogna stare attenti). L'angolo azimutale  $\gamma$  superficie-Sole è definito come:

$$\gamma = \phi - \psi \quad (\text{Eq.3.16})$$

Valori di  $\gamma$  maggiori di  $90^\circ$  o minori di  $-90^\circ$  indicano che la superficie è in ombra. Infine l'angolo tra la linea normale alla superficie irradiata e la linea Terra-Sole è chiamato angolo di incidenza  $\theta$ . E' importante nei calcoli di carichi solari su finestrature perché riguarda l'intensità della componente diretta della radiazione solare che colpisce la superficie e la capacità della superficie di assorbire, trasmettere o riflettere i raggi del sole. Il suo valore è dato da:

$$\cos \theta = \cos \beta \cos \gamma \sin \Sigma + \sin \beta \cos \Sigma \quad (\text{Eq.3.17})$$

L'irraggiamento totale  $E_t$  che raggiunge una superficie ricevente è la somma di tre componenti:

$$E_t = E_{t,b} + E_{t,d} + E_{t,r} \quad (\text{Eq.3.18})$$

dove:

$E_{t,b}$  è la componente diretta ( $\text{W/m}^2$ )

$E_{t,d}$  è la componente diffusa dal cielo ( $\text{W/m}^2$ )

$E_{t,r}$  è la componente riflessa dal terreno ( $\text{W/m}^2$ )

La componente diretta è ottenuta da una semplice relazione geometrica:

$$E_{t,b} = E_b \cos \theta \quad (\text{Eq.3.19})$$

La relazione è valida se  $\cos \theta > 0$  altrimenti  $E_{t,b} = 0$ .

La componente diffusa è più difficile da stimare a causa della natura non isotropica della radiazione diffusa: alcune parti del cielo, come il disco circumsolare o l'orizzonte, tendono ad essere più luminose rispetto al resto del cielo e ciò rende impegnativo lo sviluppo di un modello semplificato. Per superfici verticali è stato dimostrato che il rapporto  $Y$  tra la radiazione solare diffusa su una superficie verticale e quella diffusa sull'orizzonte è una semplice funzione di  $\theta$ :

$$E_{t,d} = E_d Y \quad (\text{Eq.3.20})$$

dove:

$$Y = \max(0.45, 0.55 + 0.437 \cos \theta + 0.313 \cos^2 \theta)$$

Per una superficie non verticale con pendenza  $\Sigma$ , le seguenti relazioni semplificate sono sufficienti per la maggior parte delle applicazioni:

$$E_{t,d} = E_d (Y \sin \Sigma + \cos \Sigma) \quad \text{se } \Sigma \leq 90^\circ \quad (\text{Eq.3.21})$$

$$E_{t,b} = E_d Y \sin \Sigma \quad \text{se } \Sigma > 90^\circ \quad (\text{Eq.3.22})$$

dove Y è calcolato per una superficie verticale avente lo stesso azimuth della superficie ricevente considerata.

La radiazione riflessa dal terreno, per superfici di qualsiasi orientamento, è data da:

$$E_{t,r} = E_b(\sin\beta + E_d)\rho_g \frac{1+\cos\Sigma}{2} \quad (\text{Eq.3.23})$$

dove:

$\rho_g$  è la riflettività del terreno (Fig.3.6)

Foreground Surface	Reflectance
Water (large angle of incidences)	0.07
Coniferous forest (winter)	0.07
Bituminous and gravel roof	0.13
Dry bare ground	0.2
Weathered concrete	0.22
Green grass	0.26
Dry grassland	0.2 to 0.3
Desert sand	0.4
Light building surfaces	0.6
Snow-covered surfaces:	
Typical city centre	0.2
Typical urban site	0.4
Typical rural site	0.5
Isolated rural site	0.7

Fig.3. 6 Riflettività di vari tipo di terreno

Nei nostri casi si è considerato solo la componente diretta.

### 3.2.2 I vetri e l'effetto serra

Uno degli inconvenienti più grandi di Helyx è l'assenza di un modello radiativo a bande. Questo perché il principale responsabile dell'aumento di temperatura nell'abitacolo di un'automobile è l'effetto serra. Questo fenomeno è dovuto alle particolari proprietà del vetro ed in particolare alla sua trasmissività che dipende fortemente dalla lunghezza d'onda delle radiazioni solari. Osservando Fig.3.7 si può notare che il vetro lascia passare il 90% della radiazione nel campo visibile (400-700 nm) ma risulta praticamente opaco a quella infrarossa nella regione superiore ai 3μm. Il vetro possiede, dunque, una finestra trasparente nel campo delle lunghezze d'onda comprese tra 0.3 μm e 3μm nel quale viene emesso oltre il 90% della radiazione solare. Dato che a temperatura ambiente tutta la radiazione emessa dalle superfici cade nella

regione infrarossa di uscire e, quindi, si ha un incremento. Si capisce dunque che non è possibile simulare questa dipendenza della trasmissività dalle lunghezze d'onda della radiazione solare senza un modello radiativo a bande. Per il caso stazionario si è semplicemente usato una temperatura esterna più elevata per sopperire limitatamente alla mancanza (40°).

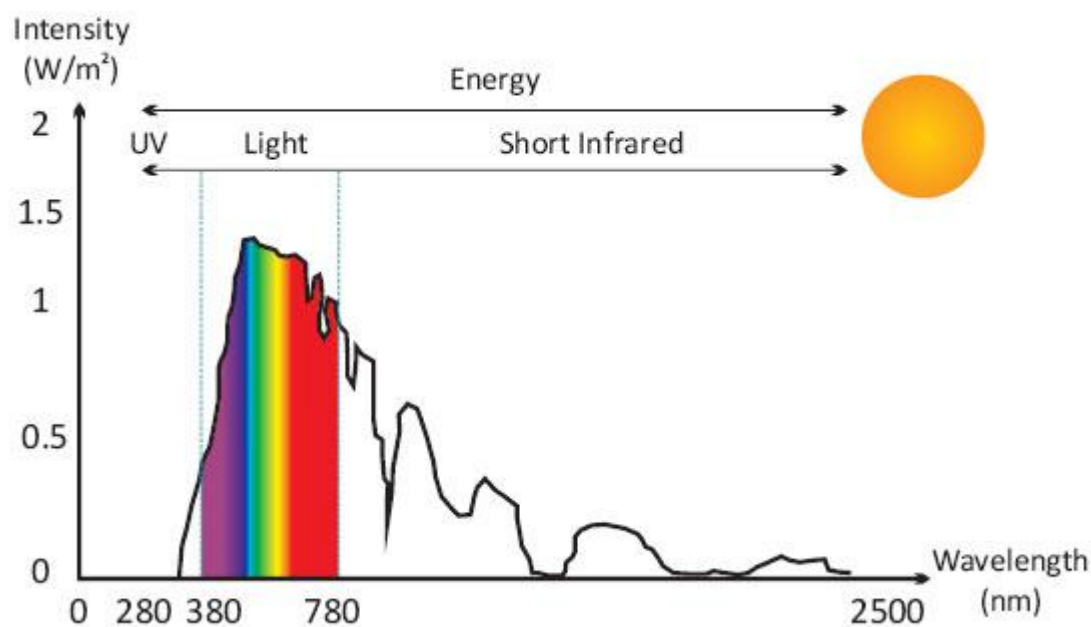


Fig.3. 7 Spettro del carico solare sulla superficie terrestre

L'altro aspetto che riguarda i vetri è che le loro proprietà radiative variano anche con l'orientamento dei raggi solari (Fig.3.8). Quando l'angolo d'incidenza aumenta da zero, la trasmittanza diminuisce, aumenta la riflessione e l'assorbimento prima aumenta, a causa dell'allungamento del percorso ottico, poi diminuisce, dato che sempre più radiazione incidente viene riflessa. Sebbene le forme delle curve siano apparentemente simili, è da notare che sia la trasmittanza che l'angolo al quale quest'ultima cambia significativamente, variano con il tipo di vetro e con lo spessore. Nel caso stazionario questo effetto verrà tenuto in conto andando a calcolare il flusso solare da imporre su ogni finestrino, sul parabrezza e sul lunotto, i quali saranno inclinati di un certo angolo.

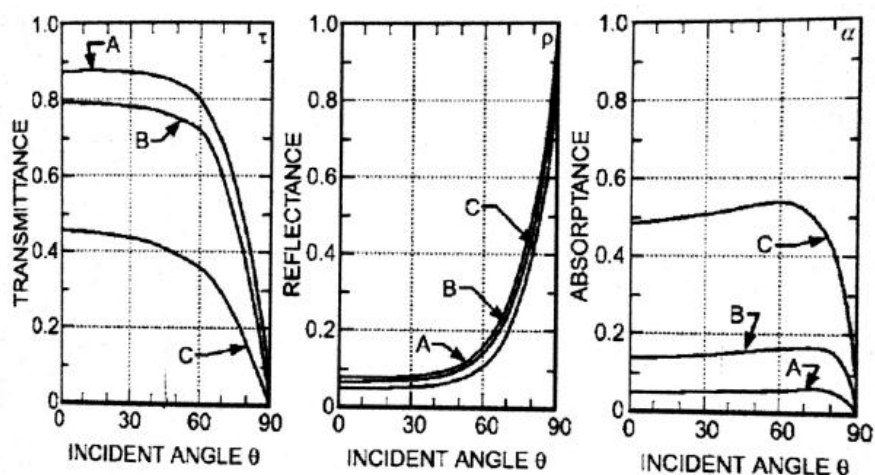


Fig.3. 8 Variazione delle proprietà radiative dei vetri con l'angolo di incidenza dei raggi solari: A per vetri a doppia resistenza, B per vetri normali e C per vetri termo-assorbenti

### 3.2.3 Soluzione del carico solare

Quanto è stato descritto nei paragrafi precedenti è stato implementato in un programma Matlab, e i risultati così ottenuti sono elencati in Tab.3.1. Gli angoli di inclinazione dei vetri utilizzati sono  $\Sigma = 45^\circ$  per i finestrini e  $\Sigma = 70^\circ$  per il parabrezza e il lunotto.

	$E_b$ [W/m <sup>2</sup> ]	$E_{t,b}$ [W/m <sup>2</sup> ]	$E_{t,d}$ [W/m <sup>2</sup> ]	$\theta$	$Y$	$\Sigma$
<i>Parabrezza</i>	1110.8	613.6477	353.3413	56.4664	0.4755	45
<i>Lunotto</i>	1110.8	925.9416	394.9487	33.5336	0.6492	45
<i>Finestrini guidatore</i>	1110.8	372.3416	290.8634	70.4157	0.5500	70
<i>Finestrini passeggero</i>	1110.8	372.3416	290.8634	70.4157	0.5500	70

Tab.3. 1 Output del solutore del carico solare ASHRAE 2009

### 3.2.4 Impostazione del solutore per il caso stazionario

Una volta generata la griglia, la finestra Case Setup nella GUI, viene resa disponibile ed è quindi possibile andare a impostare i parametri della simulazione e le condizioni al contorno.

#### *Impostazioni globali*

In Solution Modelling è possibile impostare parametri generali della simulazione (Fig.3.9):

The image shows two panels from the ANSYS Fluent GUI. The top panel is titled "Solution Modelling" and contains settings for Time, Flow, Turbulence, and Multiphase. The "Time" section has "Steady" selected. The "Flow" section has "Incompressible" selected. The "Turbulence" section has "RANS" selected, and the "Turbulence Model" dropdown is set to "Realizable k-ε". The "Multiphase" section has "VOF" unchecked. The bottom panel is titled "Thermal" and contains settings for Energy, Buoyancy, Thermal DOM Radiation, Solar Radiation, Dynamic Mesh, and Scalar Transport. "Energy" and "Buoyancy" are checked. Gravity components are 0.0, 0.0, and -9.81. "Thermal DOM Radiation" is checked, with "Divisions on θ" set to 2 and "Divisions on φ" set to 1. "Solar Radiation" is checked, with "Intensity [W/m²]" set to 800.0 and "Direction" components set to 0.198, 0.0, and -0.98. "Dynamic Mesh" has "Multibody" unchecked. "Scalar Transport" has "AoA" unchecked, "CO₂" checked, "Humidity" checked, and "Smoke" unchecked.

Panel	Section	Parameter	Value / Setting
Solution Modelling	Time	Steady	Selected
		Transient	Unselected
	Flow	Compressible	Unselected
		Incompressible	Selected
	Turbulence	RANS	Selected
		LES/DES	Unselected
Turbulence Model	Realizable k-ε	Selected	
Multiphase	VOF	Unselected	
Thermal	Energy	Energy	Checked
		Buoyancy	Checked
	Gravity [m/s²]	X	0.0
		Y	0.0
		Z	-9.81
	Thermal DOM Radiation	Thermal DOM Radiation	Checked
		Divisions on θ	2
		Divisions on φ	1
	Solar Radiation	Solar Radiation	Checked
		Intensity [W/m²]	800.0
		Direction	X: 0.198, Y: 0.0, Z: -0.98
	Dynamic Mesh	Multibody	Unselected
Scalar Transport	AoA	AoA	Unselected
		CO₂	Checked
	Humidity	Humidity	Checked
		Smoke	Unselected

Fig.3. 9 Impostazione dei parametri globali

Le impostazioni globali sono:

- Stazionario
- Incompressibile

- Modello di turbolenza RANS di tipo *Realizable*  $k-\epsilon$
- Equazione dell'energia attivata
- Radiazione termica con  $n\theta=4$  e  $n\phi=2$
- Radiazione solare con intensità  $800 \text{ W/m}^2$  e direzione  $[0.198, 0, -0.98]$
- Forza di gravità
- Trasporto scalare di  $\text{CO}_2$  e umidità

### *Materiali*

Nella sezione materiali si sceglie l'aria, impostando una temperatura di riferimento pari a  $300 \text{ K}^\circ$ .

### *Condizioni al contorno*

Le patch dell'abitacolo sono di tipo wall con le impostazioni date dalla Tab.3.2. Le portiere e il tetto sono di alluminio, il cruscotto, il volante sono di plastica e pelle nera, il tappeto e i sedili sono di pelle chiara. I finestrini sono sempre di tipo wall ma hanno i valori di trasmissività ed emissività ricavate dai grafici relativi alle proprietà del vetro (Fig.3.8) e soprattutto hanno una trasmissività solare diversa da 0, ma di un valore ricavato tramite il solutore del carico solare (ASHRAE 2009). Si suppone che la macchina sia ferma per questo caso, perciò si sono trascurate le perdite di calore attraverso le pareti (adiabatico). Si è trascurato inoltre l'emissione di calore dei manichini (metabolismo).

	<b>Momentum</b>	<b>Thermal</b>	<b>radiation</b>
F151_bocchette	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.9, \tau=0,$ $T_{\text{ext}}=313.15 \text{ K} \tau_{\text{solar}}=0$
F151_porte_plancia _e_tappeti	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.8, \tau=0,$ $T_{\text{ext}}=313.15 \text{ K} \tau_{\text{solar}}=0$
F151_tetto	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.85, \tau=0,$ $T_{\text{ext}}=313.15 \text{ K} \tau_{\text{solar}}=0$



F151_sedili_e_sedili _lateral	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.75$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$
F151_parabrezza	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.12$ , $\tau=0.73$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0.56$
F151_lunotto	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.09$ , $\tau=0.79$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0.9143$
F151_finestrini_lato _guidatore	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.24$ , $\tau=0.6$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0.279$
F151_finestrini_lato _passenger	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.24$ , $\tau=0.6$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0.279$
F151_integrazioni	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.85$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$
F151_inlet (tutte le patch)	Wall (no slip)	adiabatic	$\epsilon=0.9$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$
F151_outlet (tutte le patch)	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.85$ , $\tau=0$ , $T_{\text{ext}}=315$ $\tau_{\text{solar}}=0$
F151_tunnel	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.8$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$
F151_guidatore (tutte le patch)	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.7$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$
F151_passenger (tutte le patch)	Wall (no slip)	Heat flux $q=0$	$\epsilon=0.7$ , $\tau=0$ , $T_{\text{ext}}=313.15$ $\tau_{\text{solar}}=0$

Tab.3. 2 Condizioni al contorno di tutte le patch

Il  $\tau_{\text{solar}}$  è il rapporto tra la radiazione diretta  $E_{t,b}$  che arriva alla superficie e la radiazione del carico solare impostato su Helyx (nel nostro caso  $800 \text{ W/m}^2$ ). Mentre le altre proprietà si ricavano graficamente conoscendo l'angolo di incidenza  $\theta$ , la quale è anch'essa un'uscita del solutore del carico solare. Non vengono considerate le radiazioni diffuse. In Fig.3.10 e 3.11 si mostrano degli esempi di applicazione delle condizioni al contorno sull'interfaccia grafica di Helyx.

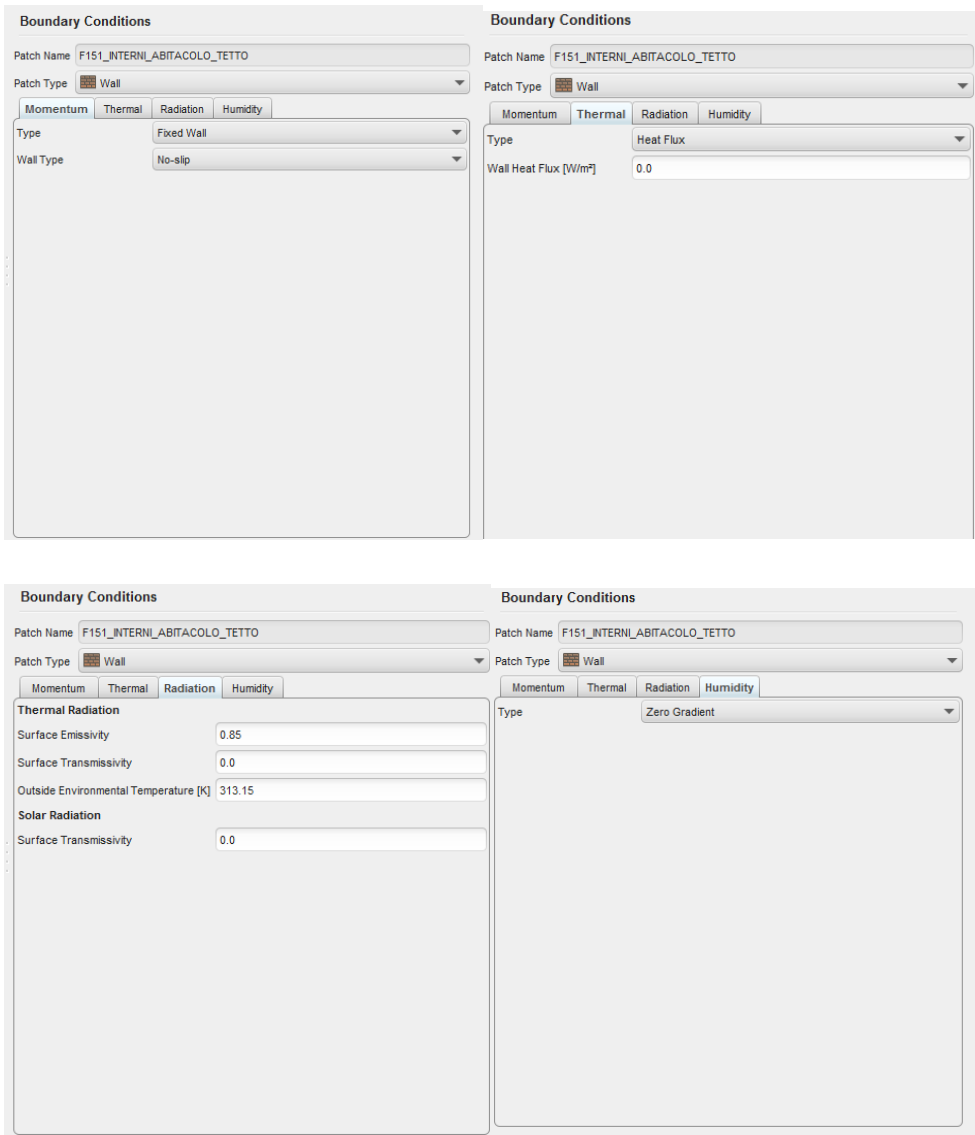


Fig.3. 10 Esempio di applicazione delle condizioni al contorno sulla GUI

**Boundary Conditions**

Patch Name: Parabrezza

Patch Type: Wall

Momentum Thermal Radiation Humidity

**Thermal Radiation**

Surface Emissivity: 0.12

Surface Transmissivity: 0.73

Outside Environmental Temperature [K]: 313.15

**Solar Radiation**

Surface Transmissivity: 0.56

Fig.3. 11 Condizioni al contorno per il Parabrezza

### Schemi numerici

Nella sezione relativa agli schemi numerici (Fig.3.12) si possono scegliere i vari metodi di discretizzazione delle variabili del termine convettivo (divergenza) e, per il laplaciano, un valore tra 0 (massima stabilità) e 1 (massima accuratezza). Questo pannello permetterà di creare il file fvScheme. Nel pannello Solver Settings si possono inserire i fattori di rilassamento (Relaxation Factors): se il valore è vicino a 1 si avrà maggiore velocità di convergenza ma minore stabilità, mentre valori vicini allo 0 comportano maggiore stabilità ma alti costi computazionali.

**Numerical Schemes**

Advection

U: Bounded Linear Upwind - 2nd Order

k: Linear Upwind - 2nd Order

epsilon: Linear Upwind - 2nd Order

T: Limited Linear - blending 1.0

w: Linear Upwind - 2nd Order

CO2: Limited Linear - blending 1.0

Laplacian

Non-orthogonal Correction: 0.333

**Solver Settings**

Solution Algorithm: SIMPLE

Non-orthogonal Correctors: 0

Residual Control

U: 0.00001

p\_rgh: 0.00001

k: 0.00001

epsilon: 0.00001

T: 0.00001

Relaxation Factors

U: 0.7

p\_rgh: 0.3

k: 0.7

epsilon: 0.7

T: 0.7

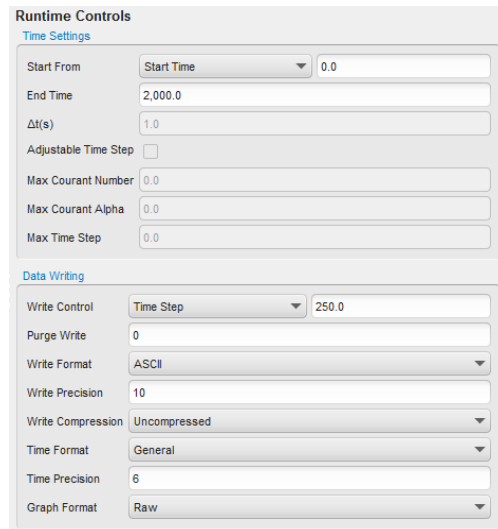
w: 0.7

CO2: 0.7

Fig.3. 12 Impostazione degli schemi di discretizzazione della soluzione

### *Controlli Run-time*

La Fig.3.13 mostra l'impostazione dei parametri per il controllo del tempo, la cui descrizione è stata fatta nel Capitolo 1. Dopo varie prove si è stabilito che con 2000 iterazioni il caso convergeva.

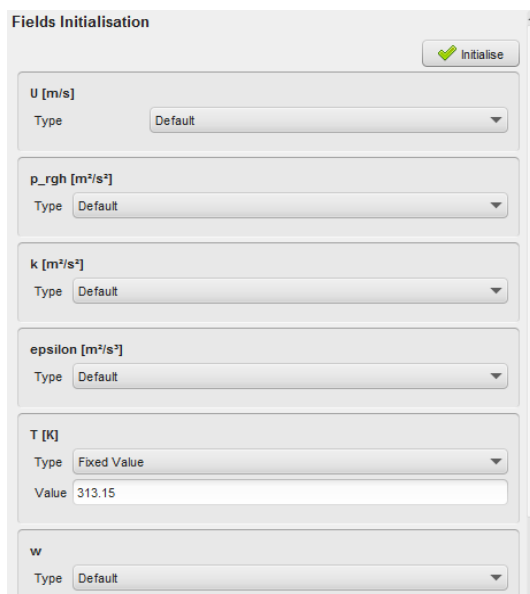


The screenshot shows the 'Runtime Controls' panel with two sections: 'Time Settings' and 'Data Writing'. In 'Time Settings', 'Start From' is set to 'Start Time' with a value of 0.0, 'End Time' is 2,000.0, and 'Δt(s)' is 1.0. Other parameters like 'Adjustable Time Step', 'Max Courant Number', 'Max Courant Alpha', and 'Max Time Step' are all set to 0.0. In 'Data Writing', 'Write Control' is 'Time Step' with a value of 250.0, 'Purge Write' is 0, 'Write Format' is 'ASCII', 'Write Precision' is 10, 'Write Compression' is 'Uncompressed', 'Time Format' is 'General', 'Time Precision' is 6, and 'Graph Format' is 'Raw'.

*Fig.3. 13 Impostazioni dei controlli Run-Time*

### *Inizializzazione dei campi*

L'inizializzazione delle variabili può essere impostata nell'apposito pannello (Fig.3.14)



The screenshot shows the 'Fields Initialisation' panel with an 'Initialise' button at the top right. It lists several variables: 'U [m/s]' with Type 'Default', 'p\_rgh [m²/s²]' with Type 'Default', 'k [m²/s²]' with Type 'Default', 'epsilon [m²/s²]' with Type 'Default', 'T [K]' with Type 'Fixed Value' and Value '313.15', and 'w' with Type 'Default'.

*Fig.3. 14 Inizializzazione delle variabili*

### *Lancio della simulazione*

Una volta completato il settaggio del solutore e delle condizioni al contorno, si procede ad avviare la simulazione nella sezione *Solver* (Fig.3.15). Cliccando sul pulsante Run, si aprirà un terminale in cui comparirà il log dell'avanzamento in tempo reale della soluzione. Si può notare che nella sezione *Solver*, il file log che verrà creato durante la simulazione viene nominato automaticamente con il nome del solutore (scelto automaticamente a seconda dei parametri che settiamo) che sta utilizzando, e in questo caso *buoyantBoussinesqSimpleFoam*.

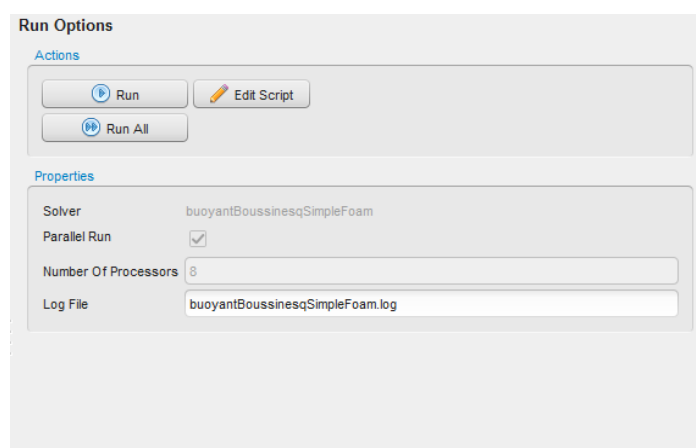


Fig.3. 15 Avvio della simulazione

## 3.3 Caso Non Stazionario

Una volta ottenuto il campo di temperature all'interno dell'abitacolo a regime in condizioni estreme di caldo con bocchette chiuse si procede con l'impostazione della soluzione non stazionaria. Le condizioni al contorno sono le stesse eccetto per la presenza stavolta di patch di inlet e di outlet. E' stato tolto il carico solare in quanto era troppo oneroso in termini di costo computazionale.

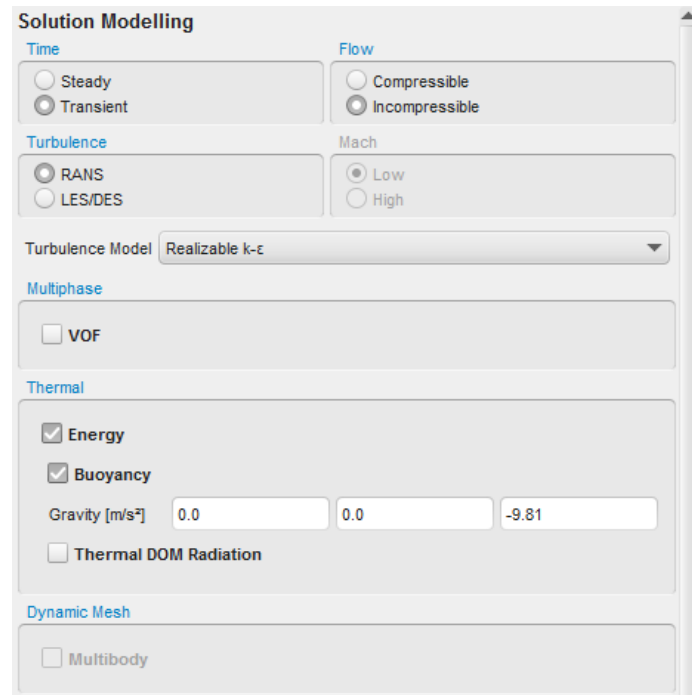
### 3.3.1 Impostazione del solutore per il caso non stazionario

#### *Impostazioni globali*

In Fig.3.16 si può notare le differenze con il caso stazionario:

- Non stazionario
- Incomprimibile

- Modello di turbolenza RANS di tipo *Realizable k- $\epsilon$*
- Equazione dell'energia attivata
- Forza di gravità



**Solution Modelling**

**Time**  
☐ Steady  
☒ Transient

**Flow**  
☐ Compressible  
☒ Incompressible

**Turbulence**  
☒ RANS  
☐ LES/DES

**Mach**  
☒ Low  
☐ High

**Turbulence Model** Realizable k- $\epsilon$

**Multiphase**  
☐ VOF

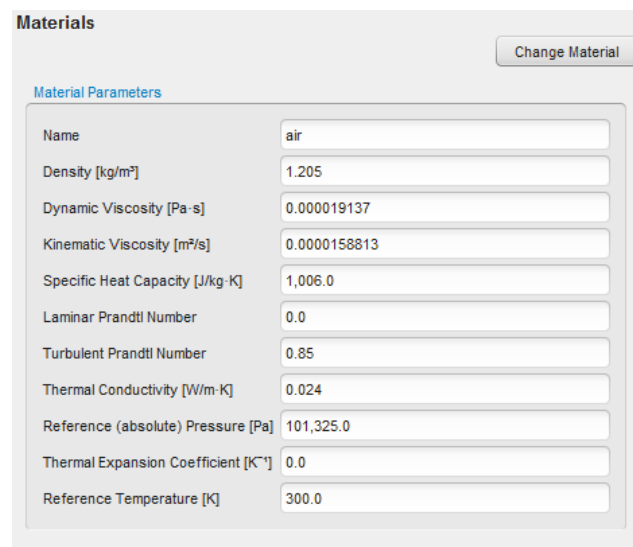
**Thermal**  
☒ Energy  
☒ Buoyancy  
 Gravity [m/s<sup>2</sup>] 0.0 0.0 -9.81  
☐ Thermal DOM Radiation

**Dynamic Mesh**  
☐ Multibody

Fig.3. 16 parametri globali per il caso non stazionario

## Materiali

Si usano le stesse impostazioni per il caso precedente: aria a quota 0 con temperatura di riferimento a 300K°.



**Materials** Change Material

**Material Parameters**

Name	air
Density [kg/m <sup>3</sup> ]	1.205
Dynamic Viscosity [Pa·s]	0.000019137
Kinematic Viscosity [m <sup>2</sup> /s]	0.0000158813
Specific Heat Capacity [J/kg·K]	1,006.0
Laminar Prandtl Number	0.0
Turbulent Prandtl Number	0.85
Thermal Conductivity [W/m·K]	0.024
Reference (absolute) Pressure [Pa]	101,325.0
Thermal Expansion Coefficient [K <sup>-1</sup> ]	0.0
Reference Temperature [K]	300.0

Fig.3. 17 Impostazione del materiale

### *Condizioni al contorno*

Le condizioni al contorno sono le stesse del caso precedente con la differenza che sono state tolte tutte le impostazioni relative al carico solare (si sta simulando in pratica un ambiente chiuso). Le bocchette d'aria d'ingresso (patch inlet) sono nella condizione di massimo con una portata d'aria totale di 400 m<sup>3</sup>/h e con il minimo di temperature. Le distribuzioni delle portate e le temperature sulle bocchette sono riassunte in Tab.3.3. Le velocità sono state calcolate in base alla velocità e su un raggio di 7 cm per bocchetta.

	<b>Distribuzione</b>	<b>Portata [m<sup>3</sup>/s]</b>	<b>Turbolenza</b>	<b>T [K°]</b>	<b>V [m/s]</b>
<b>Bocchette centrali</b>	24%	0.027	Intensity=0.05m  Mixing length=0.01m	281.9	0.59
<b>Bocchetta destra</b>	15%	0.016	Intensity=0.05m  Mixing length=0.01m	281.15	1.11
<b>Bocchetta sinistra</b>	13%	0.014	Intensity=0.05m  Mixing length=0.01m	283.85	0.96
<b>Bocchette posteriori</b>	48%	0.053	Intensity=0.05m  Mixing length=0.01m	282.95	1.7

*Tab.3. 3 Condizioni al contorno per le bocchette*

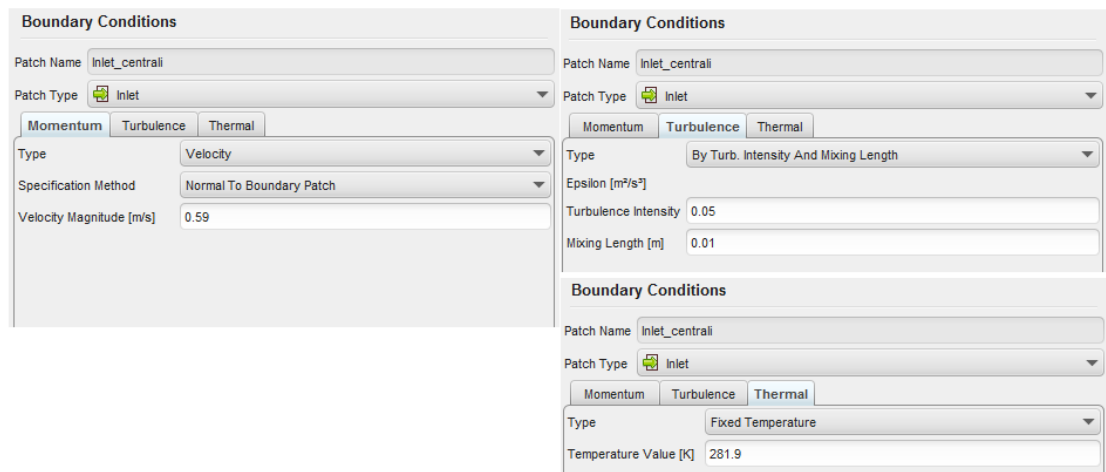


Fig.3. 18 Esempio di condizione al contorno di una bocchetta

Le uscite del flusso simulano le perdite naturali dell'abitacolo e sono necessarie qualora si intenda usare una condizione inlet sulle bocchette di ingresso. Infatti, in tal modo, l'equazione di continuità sarà soddisfatta. Le condizioni che si impongono sono mostrate nella Fig.3.19

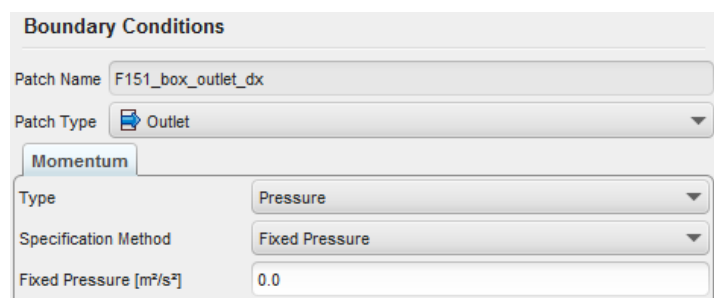


Fig.3. 19 Esempio di condizione al contorno di outlet.

Si mostrano alla fine tutte le altre condizioni al contorno in Tab.3.4. I vetri e il tetto hanno una condizione al contorno *Thermal* del tipo *convective heat flux*, questo perché essendo a contatto con l'aria esterna, subiscono un raffreddamento per convezione se la velocità relativa è diversa da 0.  $\alpha_{\text{conv}}$  è il coefficiente di scambio termico ed è proporzionale alla velocità della macchina.



	<b>Momentum</b>	<b>Thermal</b>
F151_bocchette	Wall (no slip)	Heat flux $q=0$
F151_porte_plancia _e_tappeti	Wall (no slip)	Heat flux $q=0$
F151_tetto	Wall (no slip)	Convective heat flux  $\alpha_{\text{conv}}=35 \text{ W/m}^2\text{K}$ ,  layer= thickness=3 mm, $\lambda=202.4$ W/mK
F151_sedili_e_sedili _lateralali	Wall (no slip)	Heat flux $q=0$
F151_parabrezza	Wall (no slip)	Convective heat flux  $\alpha_{\text{conv}}=35 \text{ W/m}^2\text{K}$ ,  layer= thickness=3 mm, $\lambda=0.93$ W/mK
F151_lunotto	Wall (no slip)	Convective heat flux  $\alpha_{\text{conv}}=35 \text{ W/m}^2\text{K}$ ,  layer= thickness=3 mm, $\lambda=0.93$ W/mK
F151_finestrini_lato _guidatore	Wall (no slip)	Convective heat flux  $\alpha_{\text{conv}}=35 \text{ W/m}^2\text{K}$ ,  layer= thickness=3 mm, $\lambda=0.93$ W/mK
F151_finestrini_lato _passeggero	Wall (no slip)	Convective heat flux  $\alpha_{\text{conv}}=35 \text{ W/m}^2\text{K}$ ,  layer= thickness=3 mm, $\lambda=0.93$ W/mK
F151_integrazioni	Wall (no slip)	Heat flux $q=0$
F151_tunnel	Wall (no slip)	Heat flux $q=0$

F151_guidatore (tutte le patch)	Wall (no slip)	Heat flux $q=0$
F151_passeggero (tutte le patch)	Wall (no slip)	Heat flux $q=0$

Tab.3. 4 Condizioni al contorno per il caso non stazionario

### *Schemi numerici*

Gli schemi numerici sono gli stessi del caso stazionario.

### *Controlli Run-Time*

Per un caso non stazionario non si parla più di passi iterativi, ma di time-step veri e propri. Helyx ha mostrato poche scelte in questo senso: si può scegliere un time-step fisso oppure variabile effettuando il controllo sul massimo numero di Courant. A differenza di altri software non è possibile scegliere il numero di iterazioni interne ad ogni time-step e perciò si è dovuto trovare un valore di passo temporale idoneo alla corretta convergenza della soluzione. Il valore trovato è molto basso e ciò ha comportato tempi di simulazione molto lunghi. Il time-step utilizzato è di 0.002 s e si è simulato 120 s effettivi (60000 time step). Si mostrano in Fig.3.20 le impostazioni effettuate.

The screenshot shows the 'Runtime Controls' dialog box with two main sections: 'Time Settings' and 'Data Writing'.

**Time Settings:**

- Start From: Start Time (dropdown), 0.0 (text field)
- End Time: 120.0 (text field)
- $\Delta t(s)$ : 0.002 (text field)
- Adjustable Time Step: ☐
- Max Courant Number: 20.0 (text field)
- Max Courant Alpha: 0.0 (text field)
- Max Time Step: 0.2 (text field)

**Data Writing:**

- Write Control: Time Step (dropdown), 30,000.0 (text field)
- Purge Write: 0 (text field)
- Write Format: ASCII (dropdown)
- Write Precision: 10 (text field)
- Write Compression: Uncompressed (dropdown)
- Time Format: General (dropdown)
- Time Precision: 6 (text field)
- Graph Format: Raw (dropdown)

Fig.3. 20 impostazioni Run-time per il caso non stazionario

## Funzioni

Ai fini del post-processing, è dispendioso in termini di peso dei file, salvare tutti i campi di soluzione ad ogni time-step. D'altra parte è necessario avere una corretta risoluzione nel visualizzare i risultati durante un transitorio. Si è utilizzato una particolare *function* di Helyx per salvare solo i campi di temperatura e velocità ad ogni secondo effettivo di simulazione. Questa function, *fieldWrite*, non è implementata sull'interfaccia grafica, e perciò si è dovuto compilare l'apposita sezione in *controlDict* (Fig.3.21).

```
functions
(
  Temperatura
  {
    functionObjectLibs ( "libfieldFunctionObjects.so" );
    writeControl    timeStep;
    writeInterval    500;
    spanControl      none;
    fields           ( T );
    region           region0;
    type             fieldWrite;
  }

  Velocità
  {
    functionObjectLibs ( "libfieldFunctionObjects.so" );
    writeControl    timeStep;
    writeInterval    500;
    spanControl      none;
    fields           ( U );
    region           region0;
    type             fieldWrite;
  }
)
;
```

Fig.3. 21 sezione function in controlDict per impostare fieldWrite

## Inizializzazione dei campi

L'inizializzazione dei campi va effettuata di default quando si fa il *caseSetup*, anche se non è utile al nostro caso dato che va inserito il risultato del caso stazionario.

Dopo aver settato il caso, prima di lanciarlo, si utilizza l'applicazione *parMapFields* previo inserimento del file correttamente compilato *mapFieldsDict* nella cartella *system*. Un esempio di script che compare quando si esegue *parMapFields* è mostrato in Fig.3.22.

```

      o      o      openFOAM (Engys Edition):
      o      o      version: 2.2_engysEdition-1.1
      o      o      web: http://www.engys.com
      o
    }

Build : 2.2_engysEdition-1.1-babf4b2d37c8
Exec  : parMapFields -parallel
Date  : Dec 11 2014
Time  : 20:42:40
Host  : "DANIELE-PC"
PID   : 6792
Case  : C:/Users/Daniele/Documents/newCase18.2
nProcs : 8
Slaves :
(
  "DANIELE-PC.4728"
  "DANIELE-PC.4692"
  "DANIELE-PC.4468"
  "DANIELE-PC.2136"
  "DANIELE-PC.6376"
  "DANIELE-PC.7060"
  "DANIELE-PC.4464"
)

Pstream initialized with:
  floatTransfer : 0
  nProcsSimpleSum : 0
  commType : nonBlocking
  pollingIterations : 0
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations

// *****
Create time

Target : "C:/Users/Daniele/Documents/newCase18.2/processor0"
Source : "C:/Users/Daniele/Documents/newCase18.1"

```

Fig.3. 22 Esempio di script durante l'esecuzione di parMapFields

### Lancio della simulazione

Una volta inseriti i corretti campi di temperatura e velocità della soluzione stazionario nella cartella temporale 0 del caso non stazionario, si può procedere lanciare la simulazione. Si può notare in questo caso che automaticamente viene selezionato *buoyantBoussinesqPimpleFoam* come solutore, perciò il relativo file log avrà lo stesso nome (Fig.3.23).

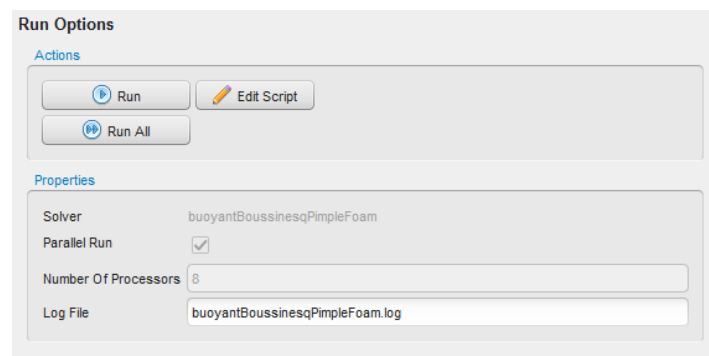


Fig.3. 23 Finestra Solver per il caso non stazionario

## Capitolo 4

# Risultati

In questo capitolo si visualizzeranno i risultati ottenuti dalle simulazioni del caso stazionario e non stazionario. Il primo è stato eseguito su una piattaforma desktop con un processore a 8 core. Il secondo è stato eseguito su un Cluster in dotazione al Dipartimento di Aerodinamica Applicata presso l'Università di Pisa e sono stati utilizzati 64 processori.

### 4.1 Risultati caso stazionario

Il caso stazionario è stato eseguito in circa 20 ore. Di seguito (Fig.4.1) si mostra l'andamento dei residui che mostrano una buona convergenza della soluzione:

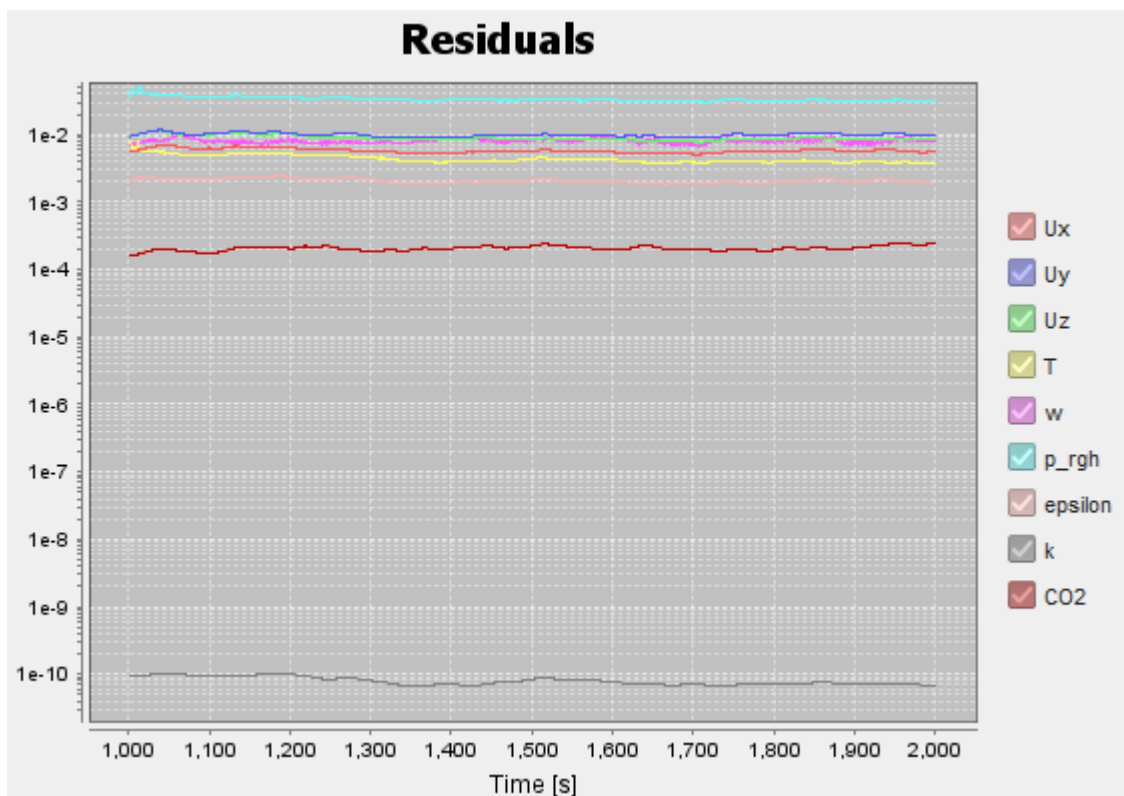
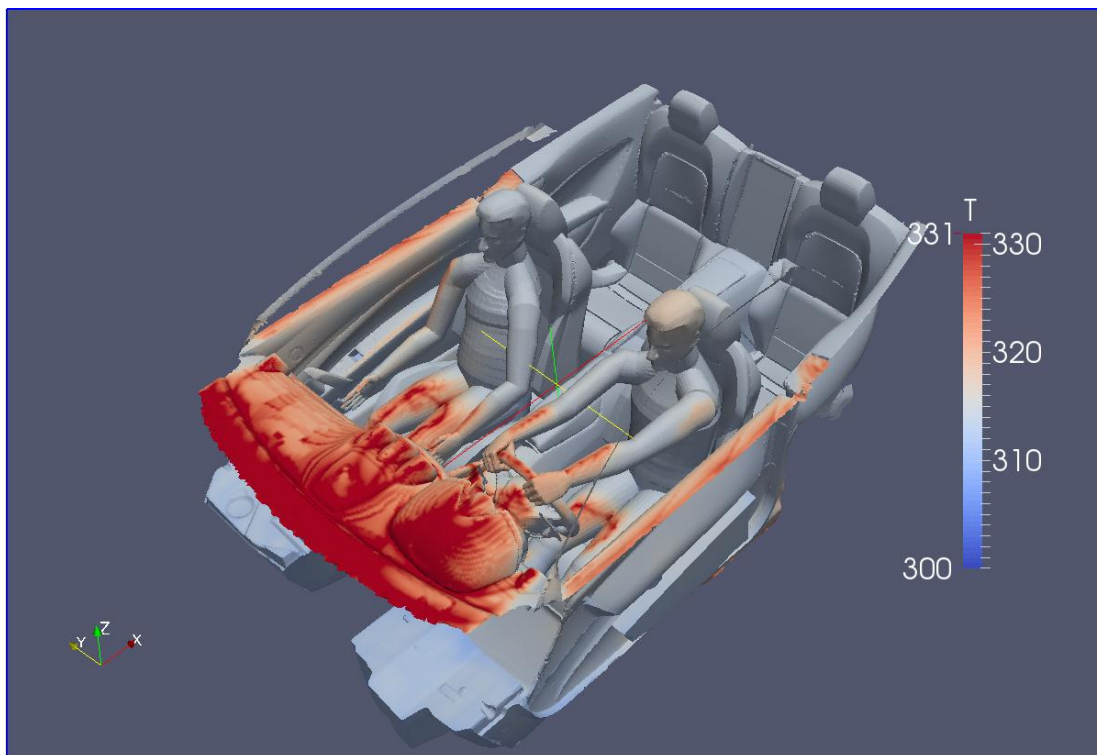
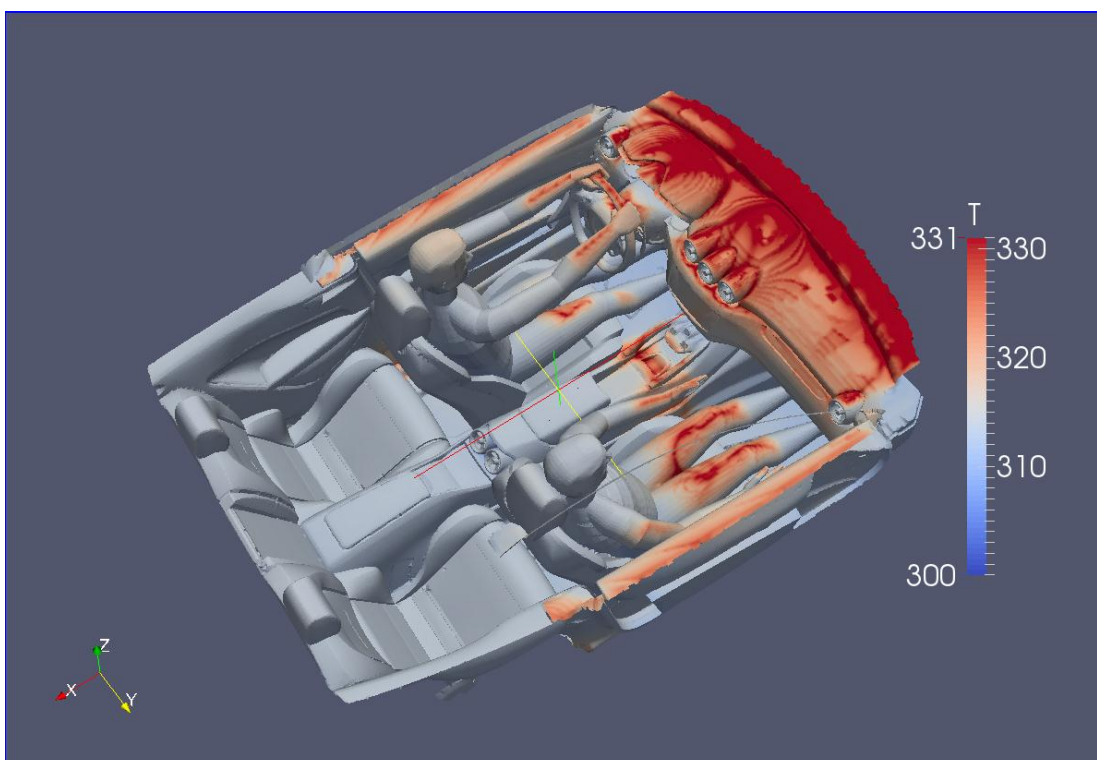


Fig.4. 1 Andamento dei residui

I risultati sono stati visualizzati tramite il software open source Paraview. Si mostrano alcune visualizzazioni.

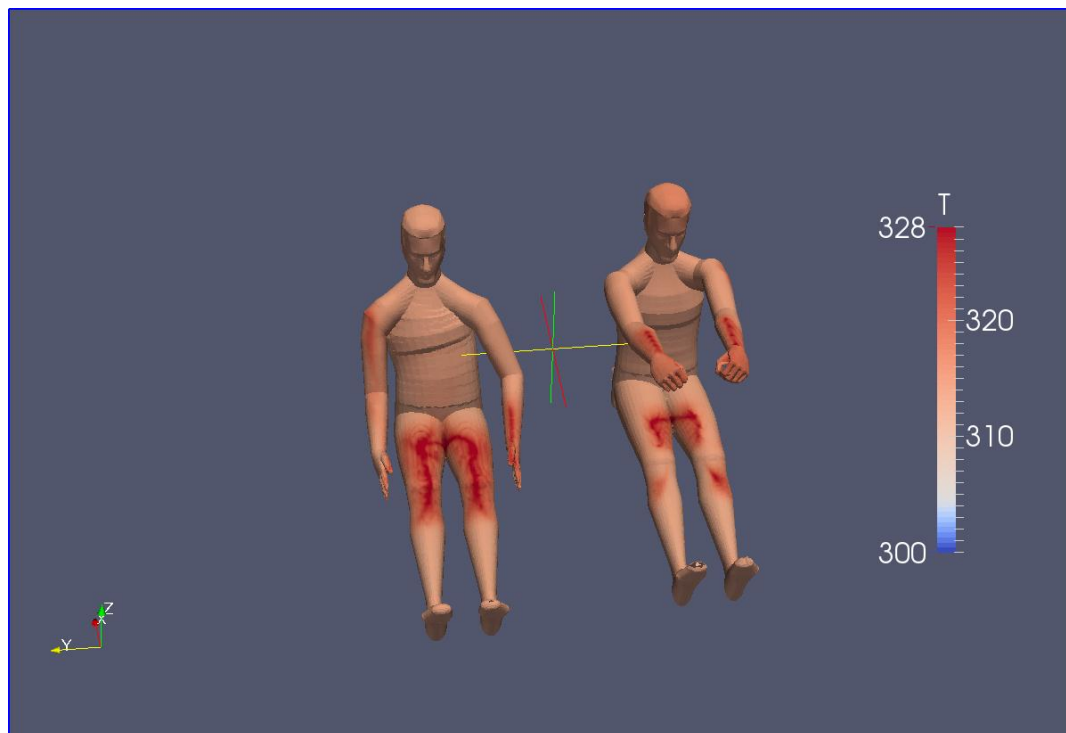


*Fig.4. 2 Vista frontale della visualizzazione*

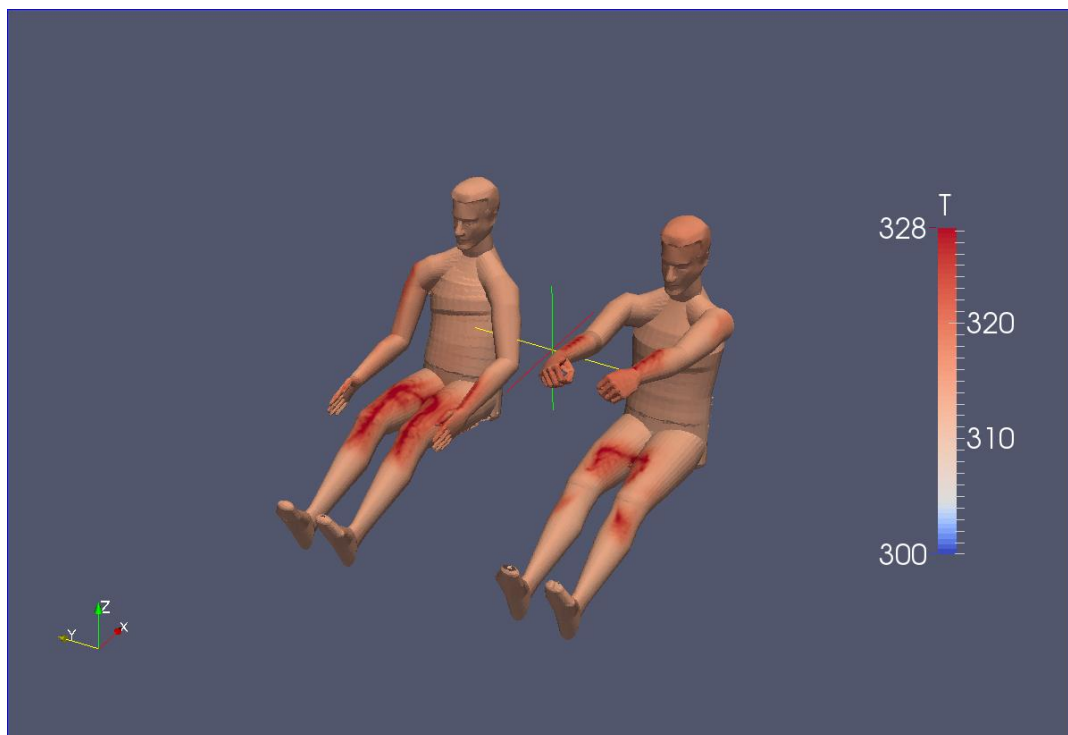


*Fig.4. 3 Vista dall'alto della visualizzazione*

Si nota che le temperature massime possono raggiungere i 331 °K, ovvero i 58° Celsius in alcune zone sul cruscotto (Fig.4.1 e Fig.4.2). Mediamente si può vedere che sulle superfici la temperatura media è pari a quella esterna (si è trascurato l'effetto serra).



*Fig.4. 4 Vista 1 dei manichini*



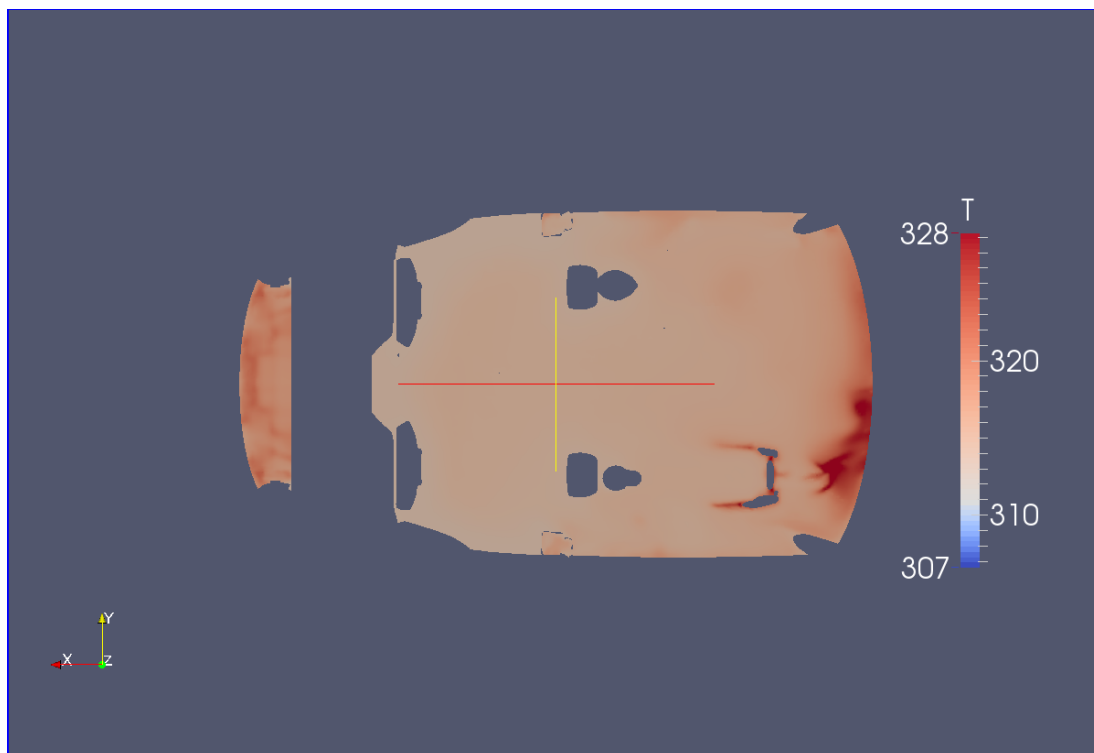
*Fig.4. 5 Vista 2 dei manichini*

Se isoliamo i manichini possiamo notare dei picchi massimi di temperatura a 328.25 °K, ovvero a circa 55° Celsius sulle cosce, sulla spalla del passeggero e sugli avambracci del guidatore (Fig.4.4 e Fig.4.5).

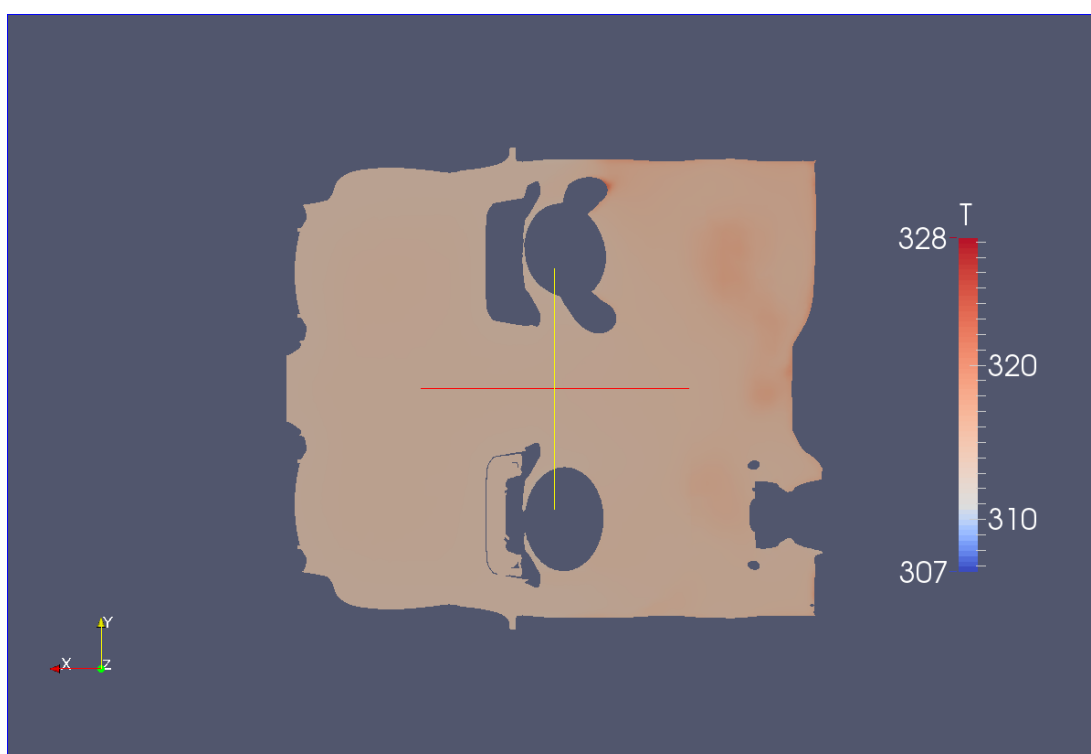


*Fig.4. 6 Sezione verticale che interseca gambe e braccia*





*Fig.4. 7 Sezione orizzontale altezza testa*



*Fig.4. 8 Sezione orizzontale altezza busto*

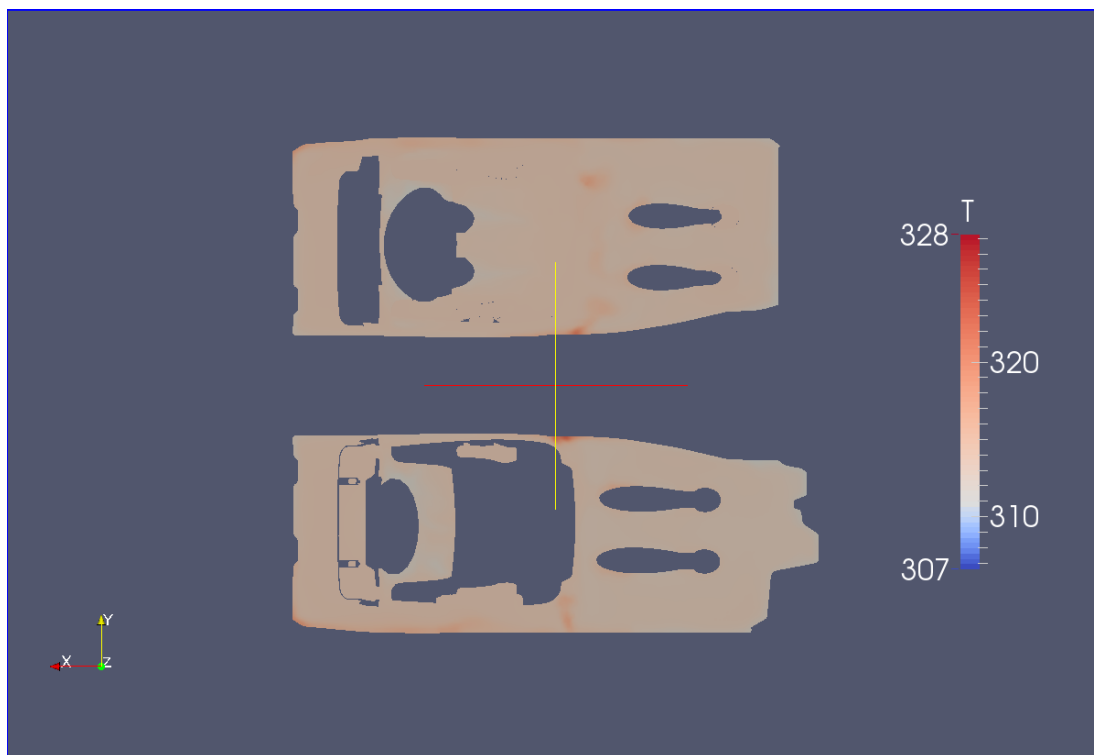


Fig.4. 9 Sezione orizzontale altezza piedi

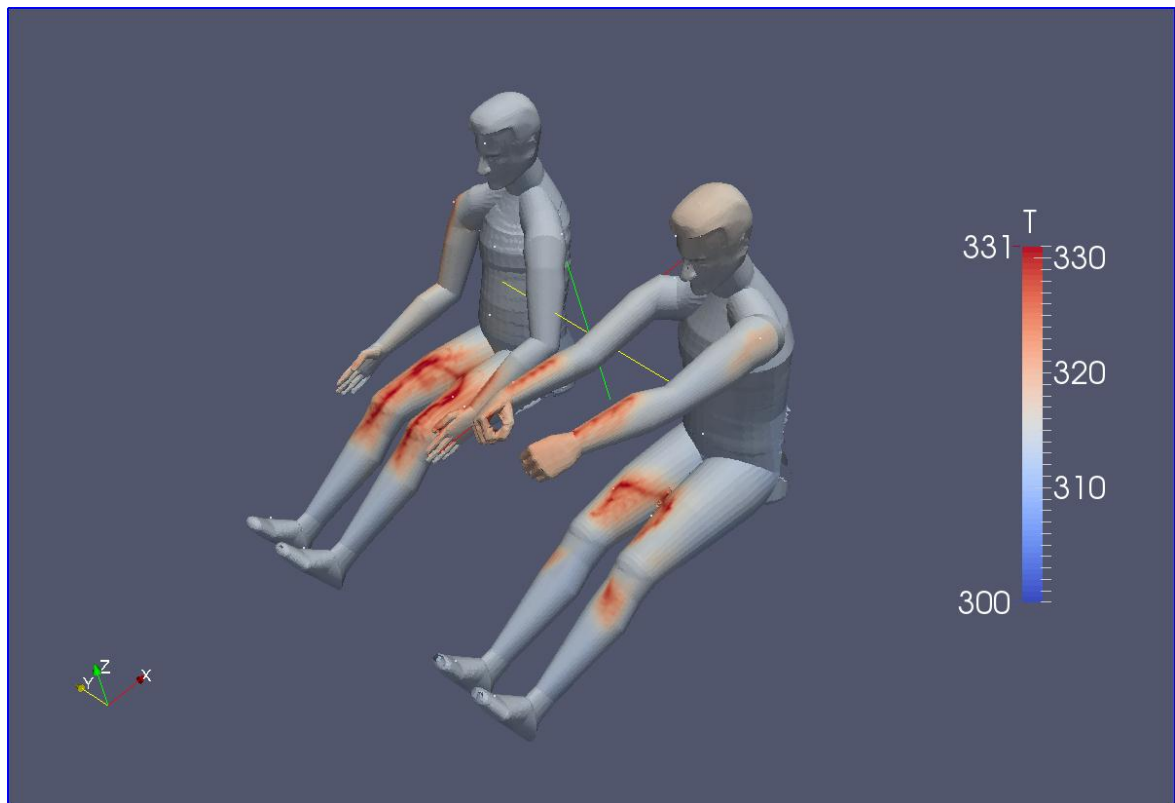
Nelle Fig.4.6, 4.7, 4.8 e 4.9 si mostrano alcune sezioni del campo di temperatura all'interno dell'abitacolo. Si può notare che la temperatura è abbastanza uniforme nelle parti basse della macchina, per poi diventare più eterogenea quando incontra le superfici colpite dal sole. Inoltre si nota come l'aria calda sale. Le temperature massime dell'aria sono dove entrano a contatto con le superfici più calde, ovvero sulle braccia, sulle cosce dei manichini e sul cruscotto.

Si è inoltre introdotto delle sonde all'interno dell'abitacolo per monitorare le temperature (Fig.4.10 e Tab.4.1):

Posizione Sonda	Temperatura [°K]
Passeggero piede sinistro	313.202
Passeggero piede destro	313.208
Passeggero coscia destra	326.077

Passeggero coscia sinistra	322.904
Passeggero mano sinistra	317.668
Passeggero mano destra	312.801
Passeggero spalla destra	322.121
Passeggero spalla sinistra	313.445
Passeggero petto	313.547
Passeggero addome	314.578
Passeggero testa	313.908
Guidatore piede destro	313.205
Guidatore piede sinistro	313.264
Guidatore mano sinistra	325.02
Guidatore mano destra	321.899
Guidatore coscia destra	322.067
Guidatore coscia sinistra	316.989
Guidatore spalla sinistra	316.245
Guidatore spalla destra	313.504
Guidatore testa	316.839
Guidatore addome	313.401
Guidatore petto	313.594

*Tab.4. 1 Sonde di temperatura*



*Fig.4. 10 Posizioni delle sonde*

Coerentemente con le visualizzazioni i punti più caldi sono le cosce, le spalle rivolte verso il finestrino e gli avanbracci-mani.

## 4.2 Risultati caso non stazionario

I risultati del caso non stazionario sono stati ottenuti da una simulazione durata circa 216 ore per ottenere 120 secondi di tempo effettivo. Sono stati usati 64 processori. Il time step utilizzato per la simulazione è di 2 ms, mentre è stata fatta l'uscita dei campi di temperature e velocità ogni secondo di simulazione. In Fig.4.11 e Fig.4.12 si mostrano una sezione orizzontale all'altezza delle bocchette dopo il primo secondo di simulazione rispettivamente del campo di temperature e del campo di velocità.

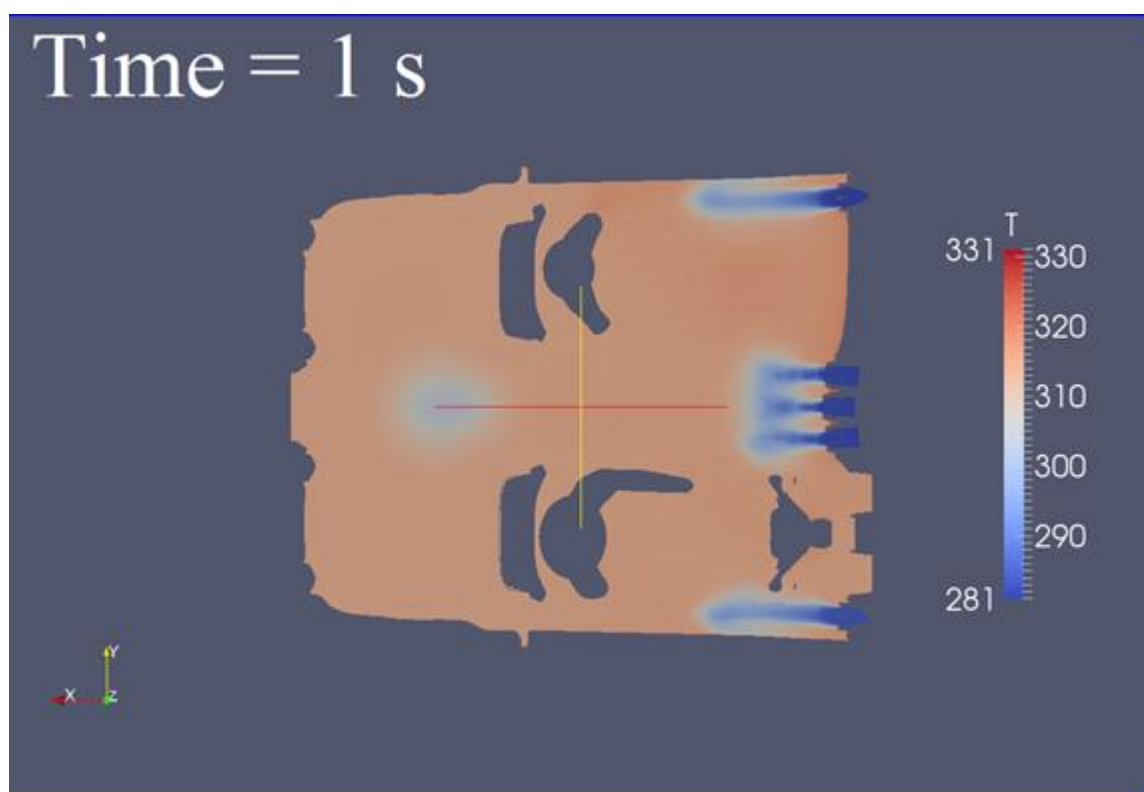


Fig.4. 11 Sezione orizzontale del campo di temperature all'altezza delle bocchette ( $t=1\text{s}$ )

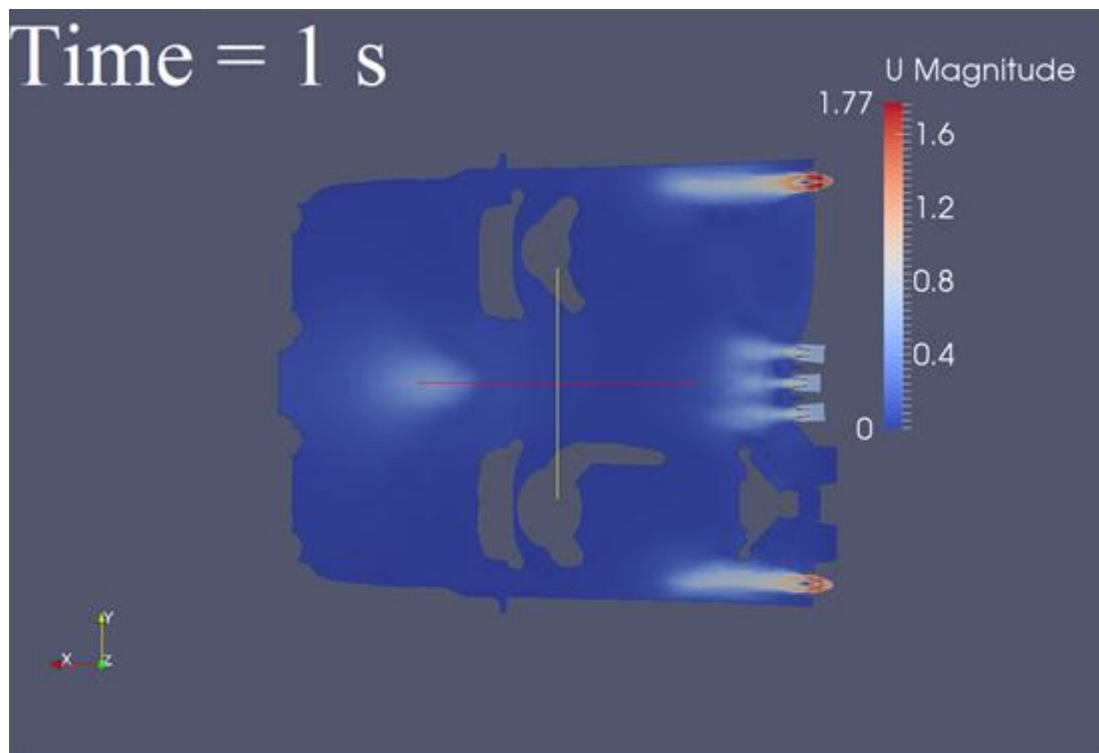


Fig.4. 12 Sezione orizzontale dei campi di velocità (modulo) all'altezza delle bocchette ( $t=1s$ )

In Fig.4.13 e 4.14 si mostrano una sezione verticale dei campi di temperatura e velocità, rispettivamente, in corrispondenza delle bocchette posteriori sempre al primo secondo di simulazione.

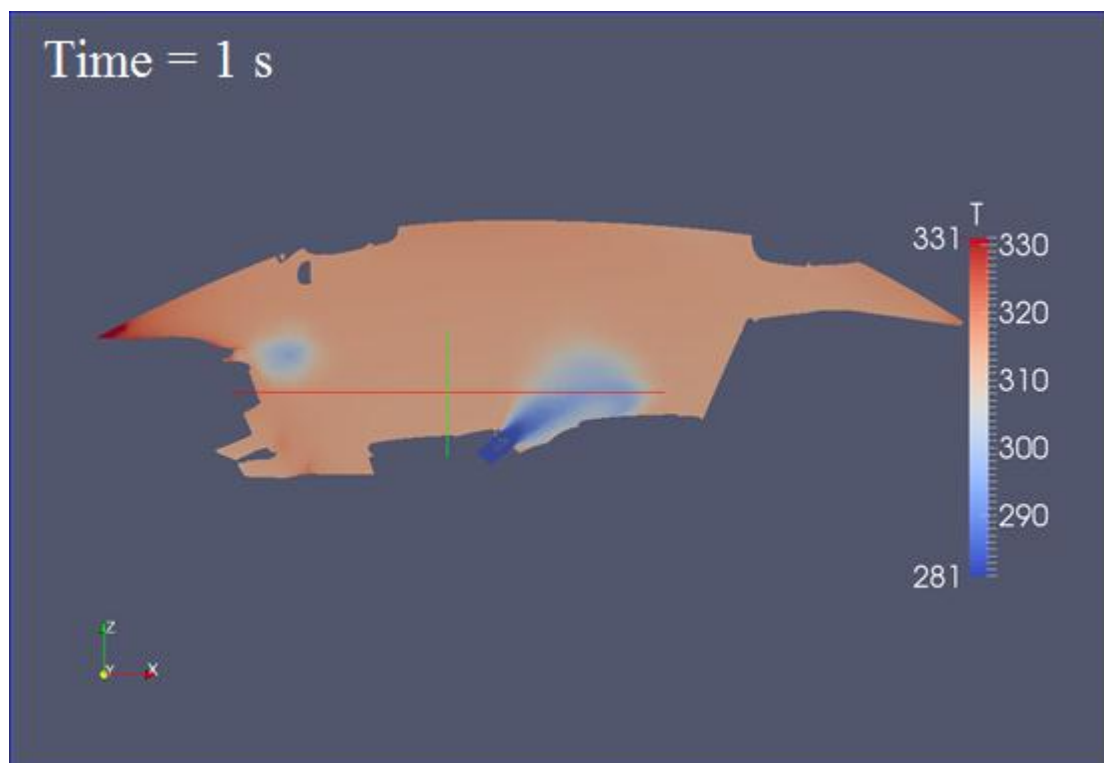
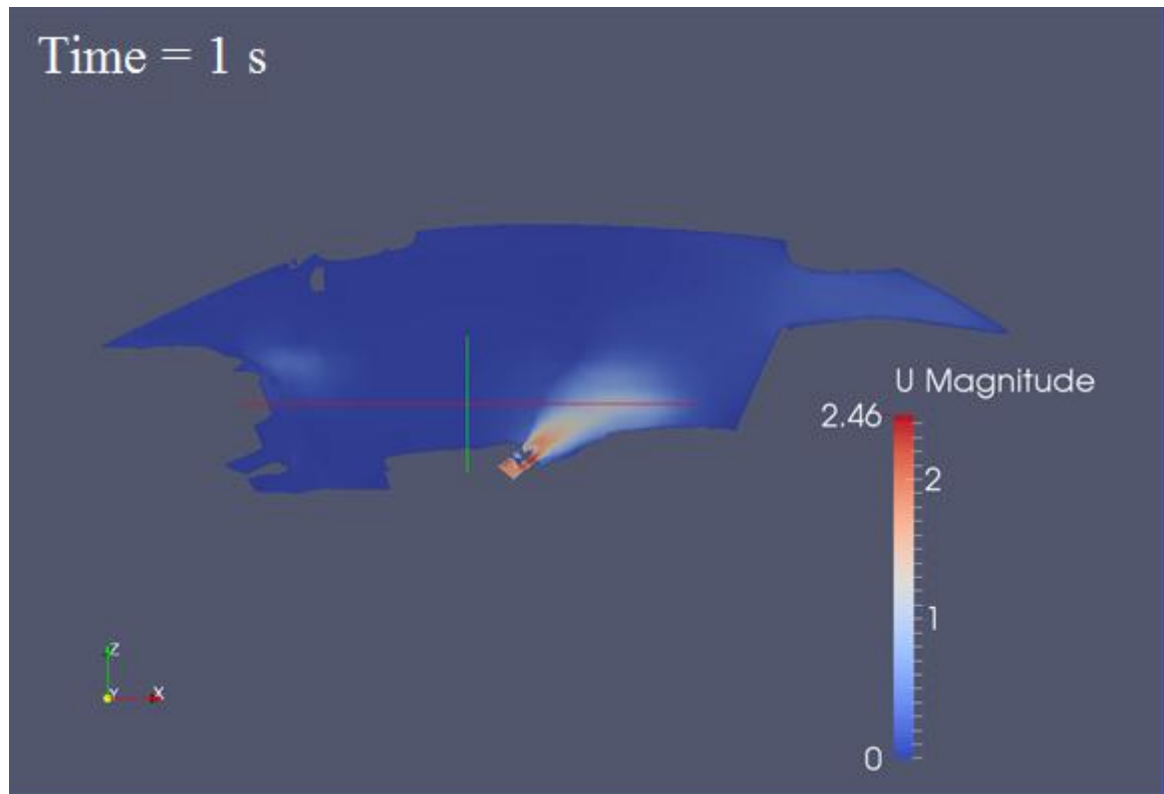


Fig.4. 13 Sezione verticale del campo di temperatura in corrispondenza delle bocchette posteriori ( $t=1s$ )



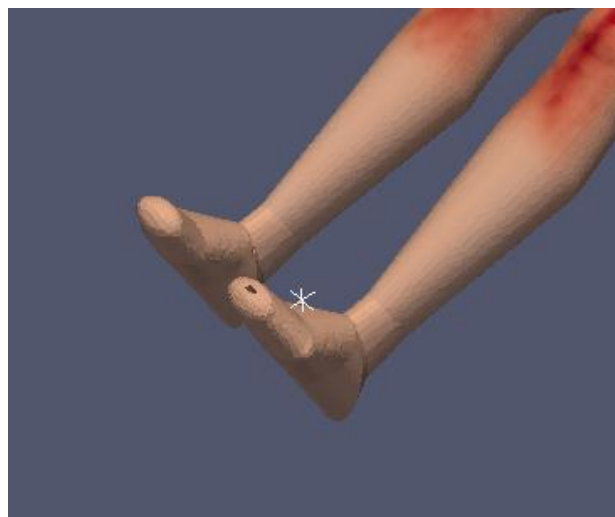
*Fig.4. 14 Sezione verticale del campo di velocità (modulo) in corrispondenza delle bocchette posteriori (t=1s)*

Per tutto il transitorio sono disponibili dei video con varie sezioni che mostrano l'andamento del campo di temperature al variare del tempo. I dati sono poi disponibili per ulteriori analisi punto per punto per istante (ogni secondo).

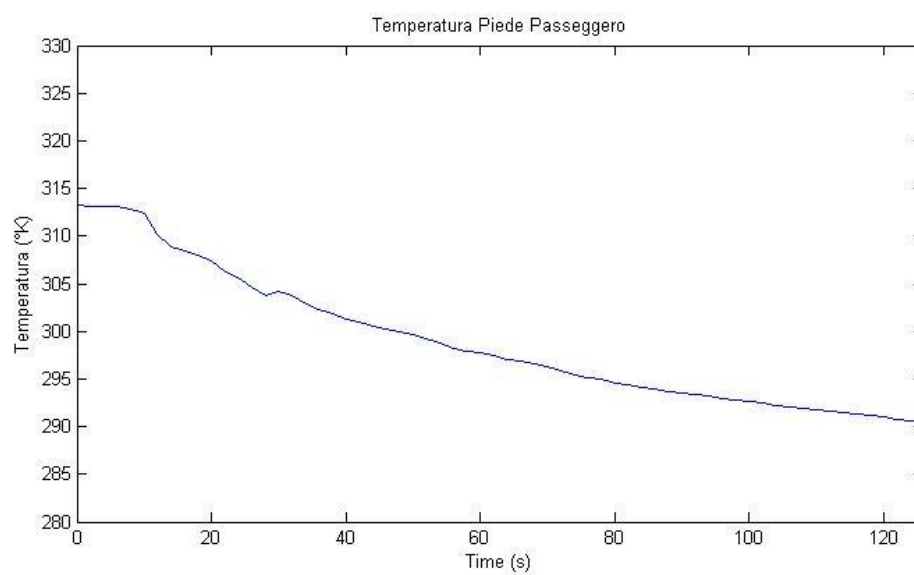
#### **4.2.1 Risultati sonde**

Si mostrano in seguito i grafici relativi alle uscite delle sonde al variare del tempo:

## *Piede Passeggero*



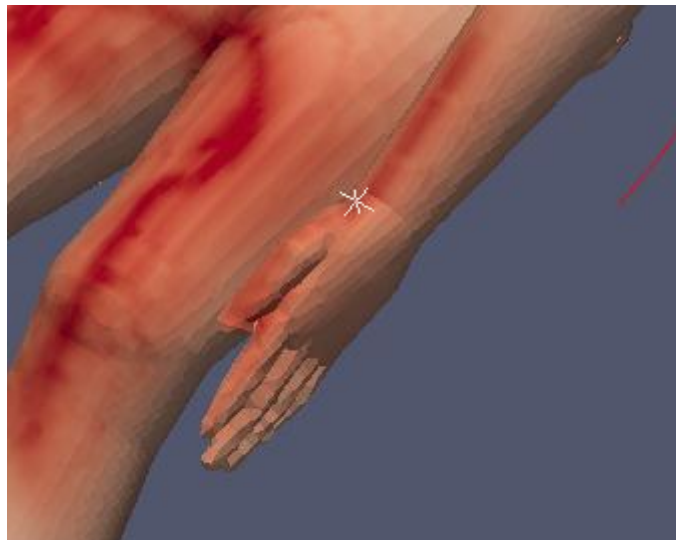
*Fig.4. 15 Sonda posizionata sul piede sinistro del passeggero*



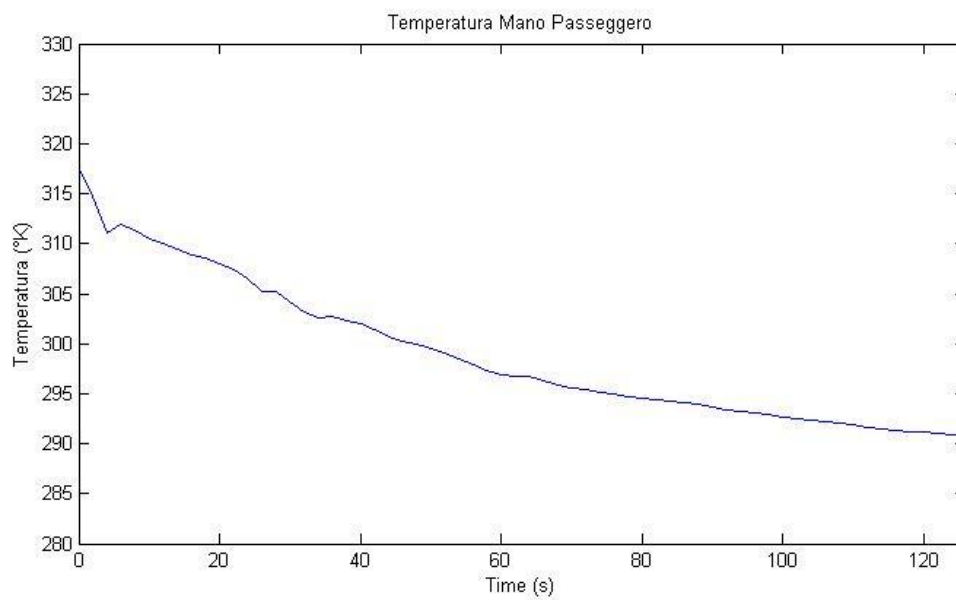
*Fig.4. 16 Risultati della sonda posta sul piede sinistro del passeggero*



## *Mano Passeggero*



*Fig.4. 17 Sonda posizionata sulla mano sinistra del passeggero*

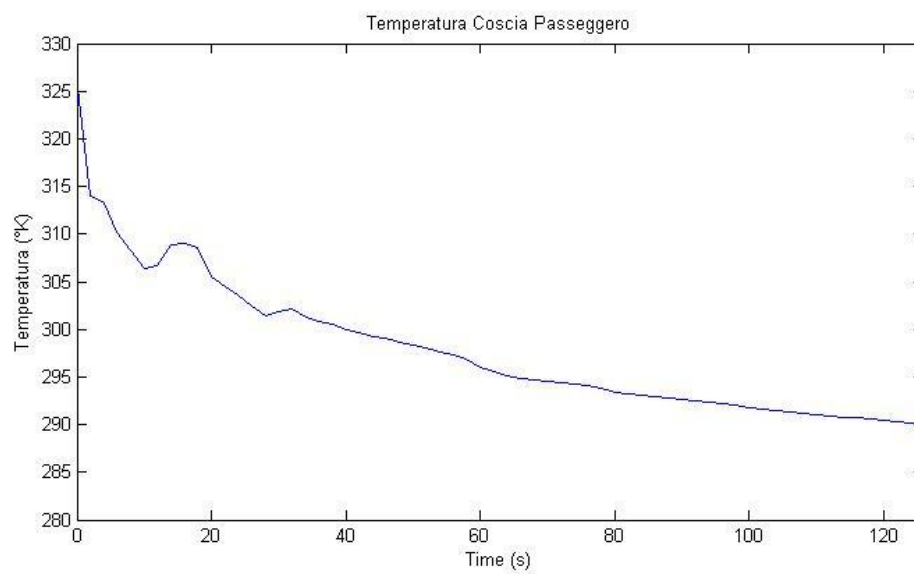


*Fig.4. 18 Risultati della sonda posta sulla mano sinistra del passeggero*

## *Coscia Passeggero*

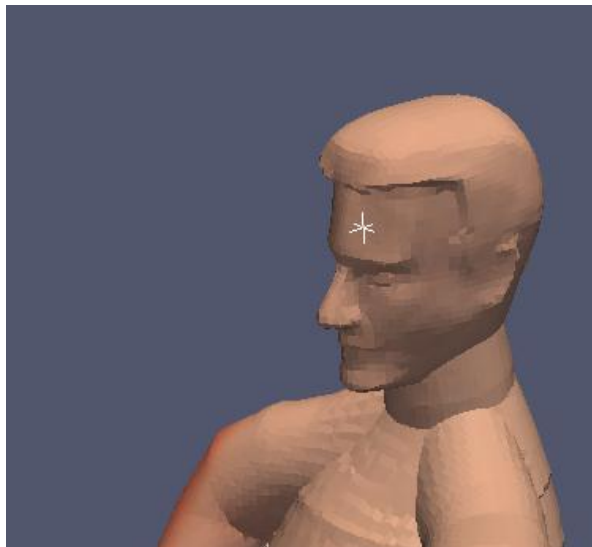


*Fig.4. 19 Sonda posizionata sulla coscia destra del passeggero*

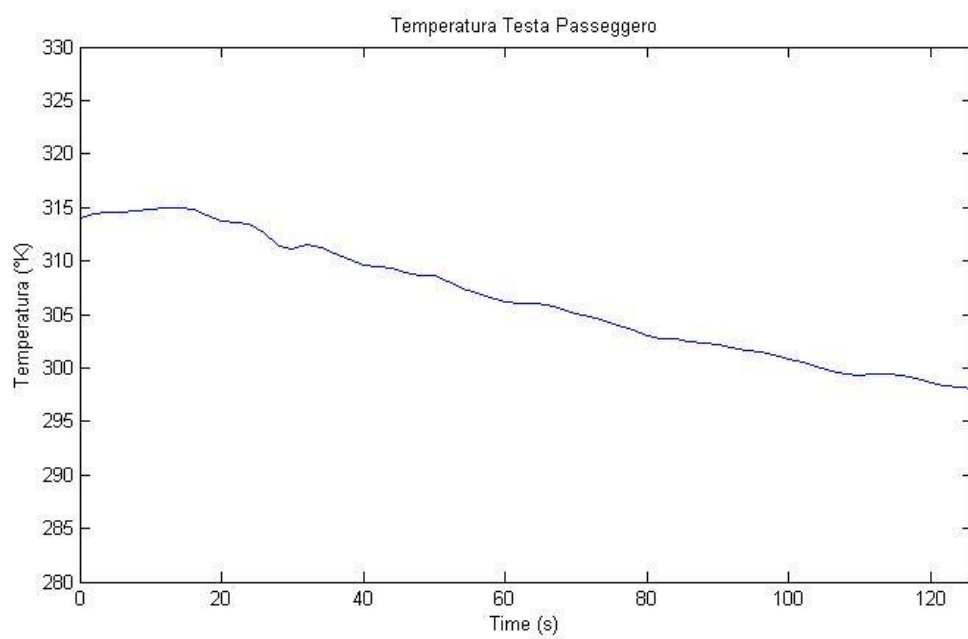


*Fig.4. 20 Risultati della sonda posizionata sulla coscia destra del passeggero*

## *Testa Passeggero*



*Fig.4. 21 Sonda posizionata sulla fronte del passeggero*

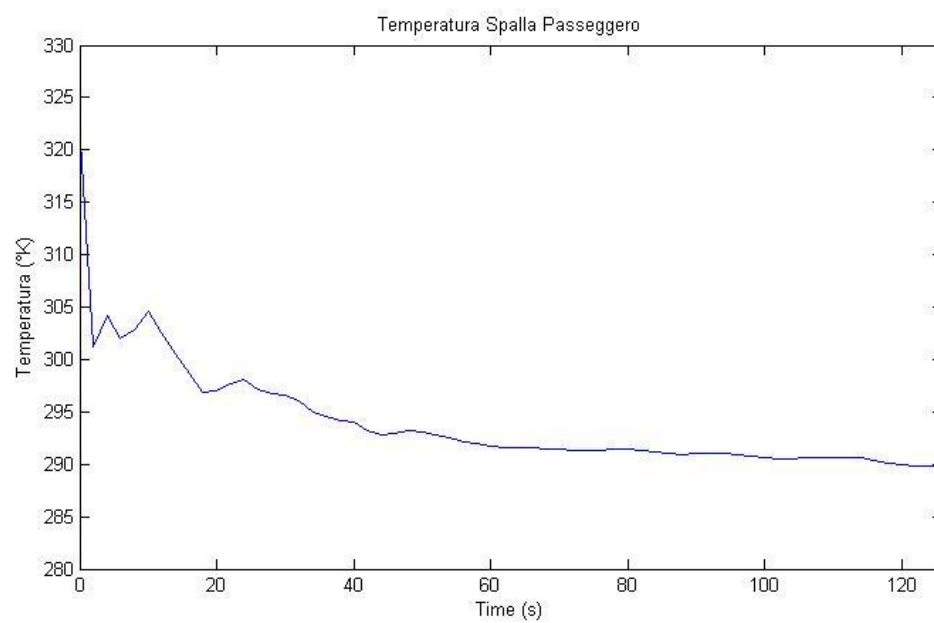


*Fig.4. 22 Risultati della sonda posizionata sulla fronte del passeggero*

## *Spalla Passeggero*

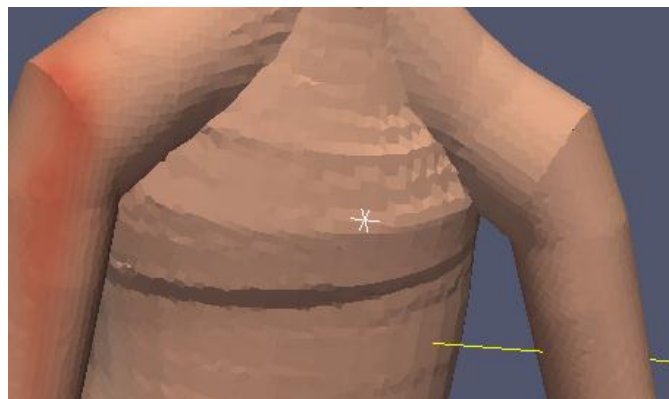


*Fig.4. 23 Sonda posizionata sulla spalla destra del passeggero*

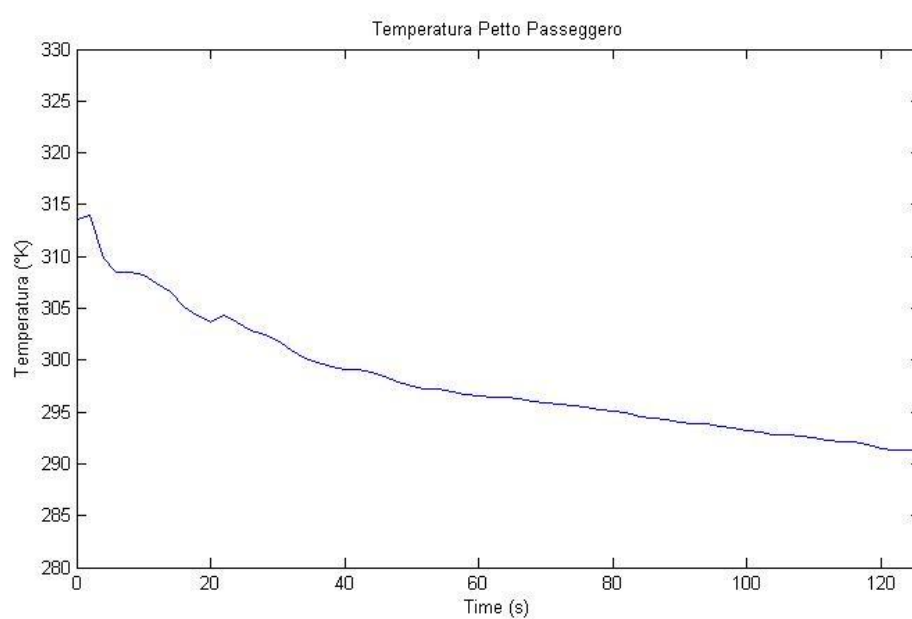


*Fig.4. 24 Risultati della sonda posta sulla spalla destra del passeggero*

## *Petto Passeggero*



*Fig.4. 25 Sonda posizionata sul petto del passeggero*

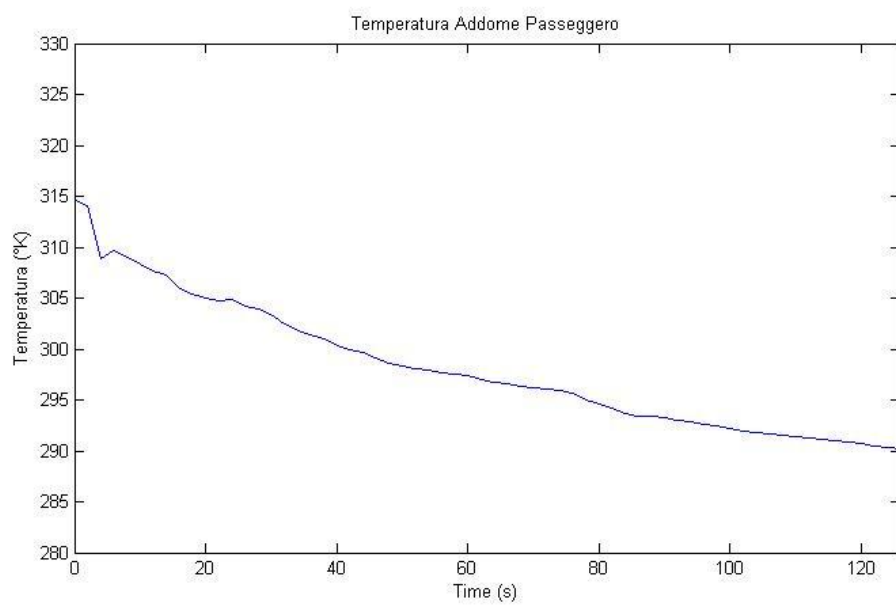


*Fig.4. 26 Risultati della sonda posizionata sul petto del passeggero*

## *Addome Passeggero*



*Fig.4. 27 Sona posizionata sull'addome del passeggero*

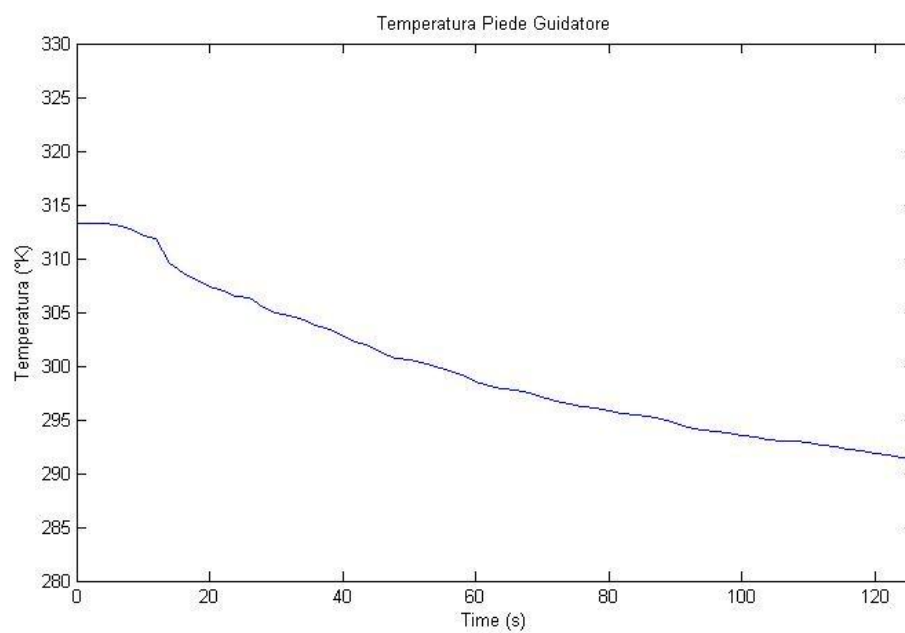


*Fig.4. 28 Risultati della sonda posizionata sull'addome del passeggero*

## *Piede Guidatore*



*Fig.4. 29 Sonda posizionata sul piede destro del guidatore*

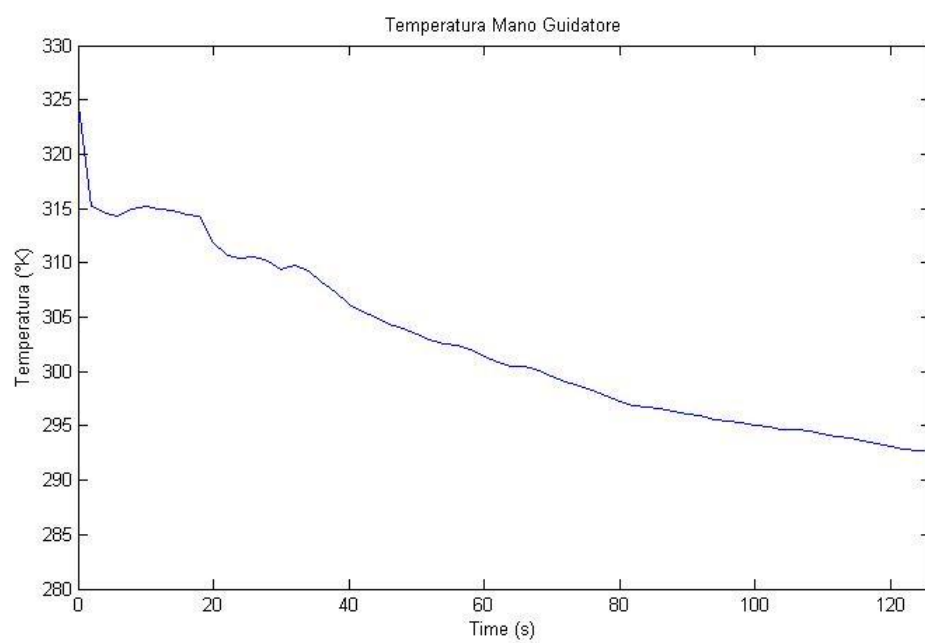


*Fig.4. 30 Risultati della sonda posta sul piede destro del guidatore*

## *Mano Guidatore*



*Fig.4. 31 Sonda posizionata posta sulla mano sinistra del guidatore*



*Fig.4. 32 Risultati della sonda posta sulla mano sinistra del guidatore*



## Testa Guidatore



Fig.4. 33 Sonda posta sulla fronte del guidatore

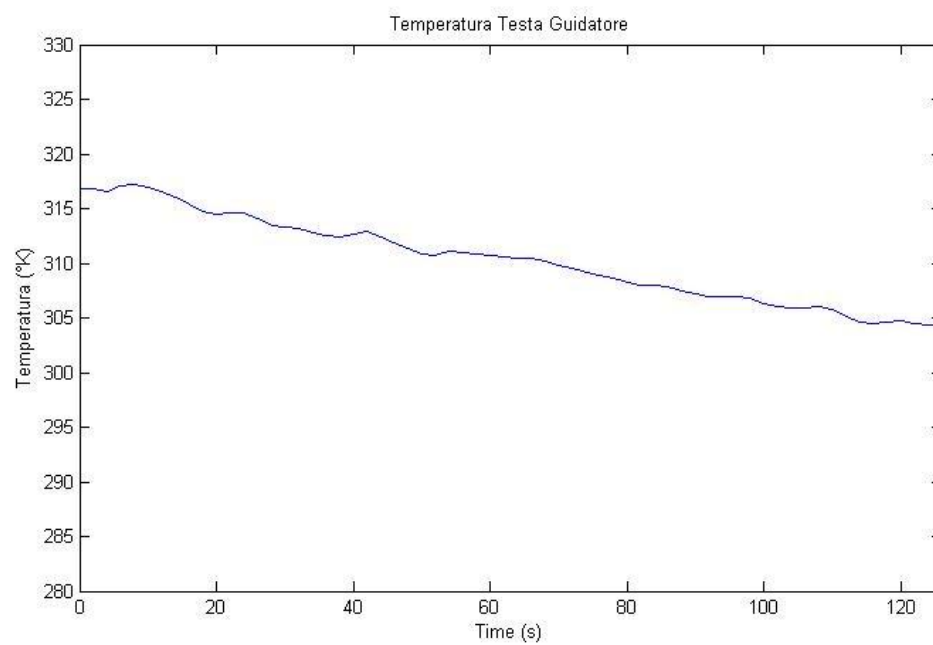
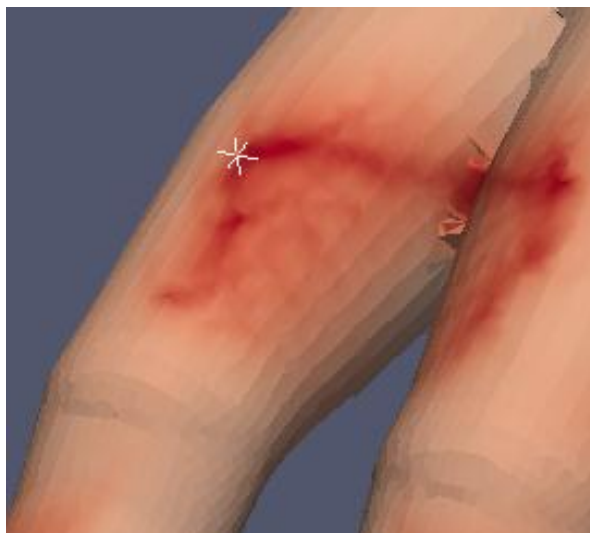
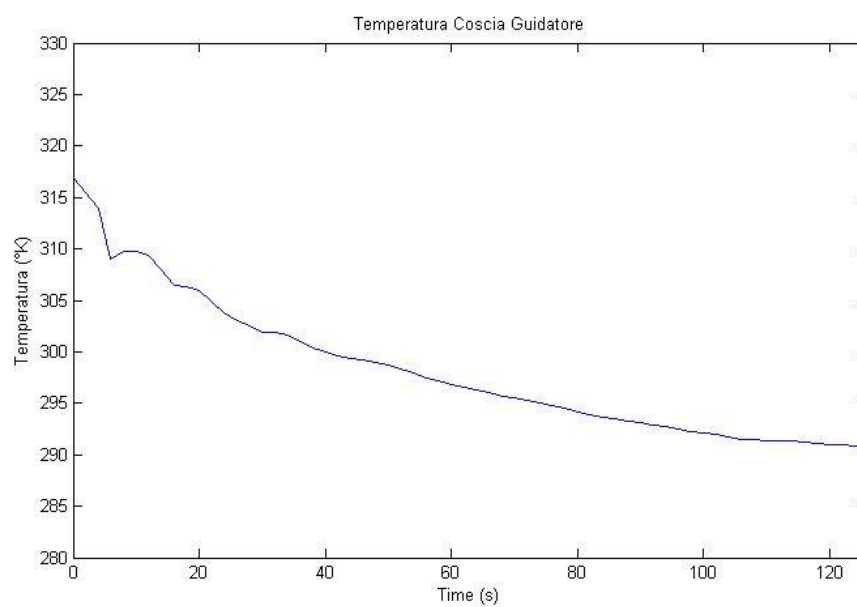


Fig.4. 34 Risultati della sonda posta sulla fronte del guidatore

## *Coscia Guida*



*Fig.4. 35 Sonda posizionata sulla coscia destra del guidatore*

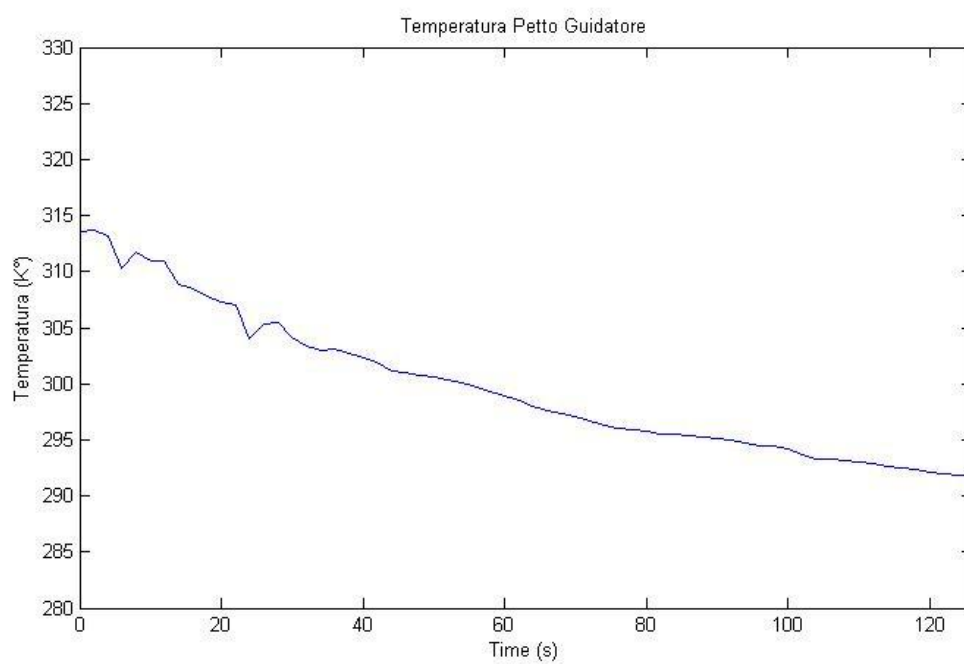


*Fig.4. 36 Risultati della sonda posizionata sulla coscia destra del guidatore*

## *Petto Guida*



*Fig.4. 37 Sonda posizionata sul petto del guidatore*

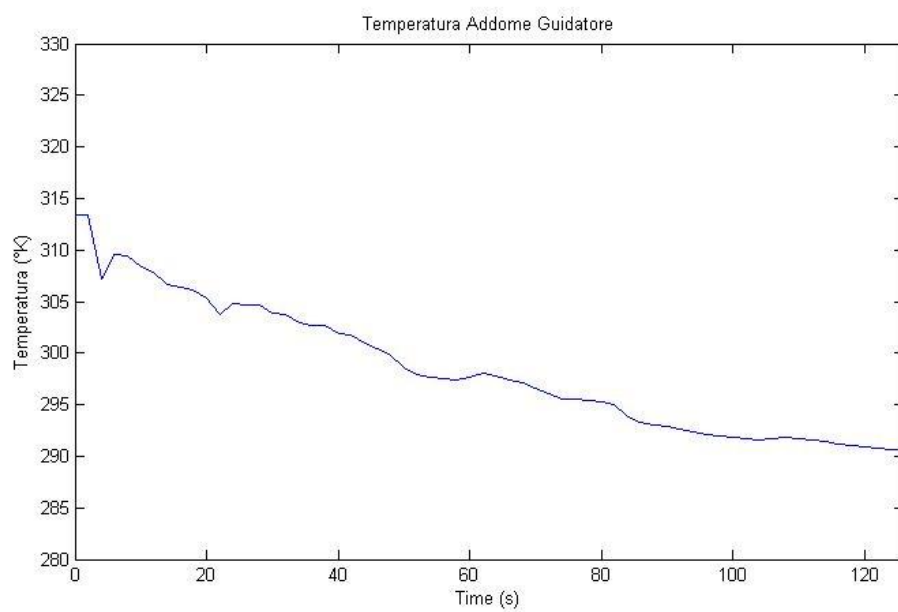


*Fig.4. 38 Risultati della sonda posta sul petto del guidatore*

## *Addome Guidatore*



*Fig.4. 39 Sonda posizionata sull'addome del guidatore*



*Fig.4. 40 Risultati della sonda posizionata sull'addome del guidatore*

Dai risultati delle sonde si può notare che vi sono delle oscillazioni durante il transitorio e ciò è dovuto al movimento dell'aria. Inoltre si nota che le zone più vicine al getto dell'aria condizionata subiscono un raffreddamento più brusco (per esempio la spalla destra del passeggero) anche se poi si stabilizzano, mentre le zone lontane si raffreddano in modo più graduale (per esempio i piedi).

#### 4.2.2 Ultimo istante

Infine si analizza l'ultimo tempo della simulazione a 127 secondi. Ovviamente non siamo ancora a regime, ma si può già mostrare che le temperature sono decisamente più gradevoli per il comfort dei passeggeri. Si mostrano in Fig.4.41-44 le stesse sezioni precedentemente viste per  $t=127$  s.

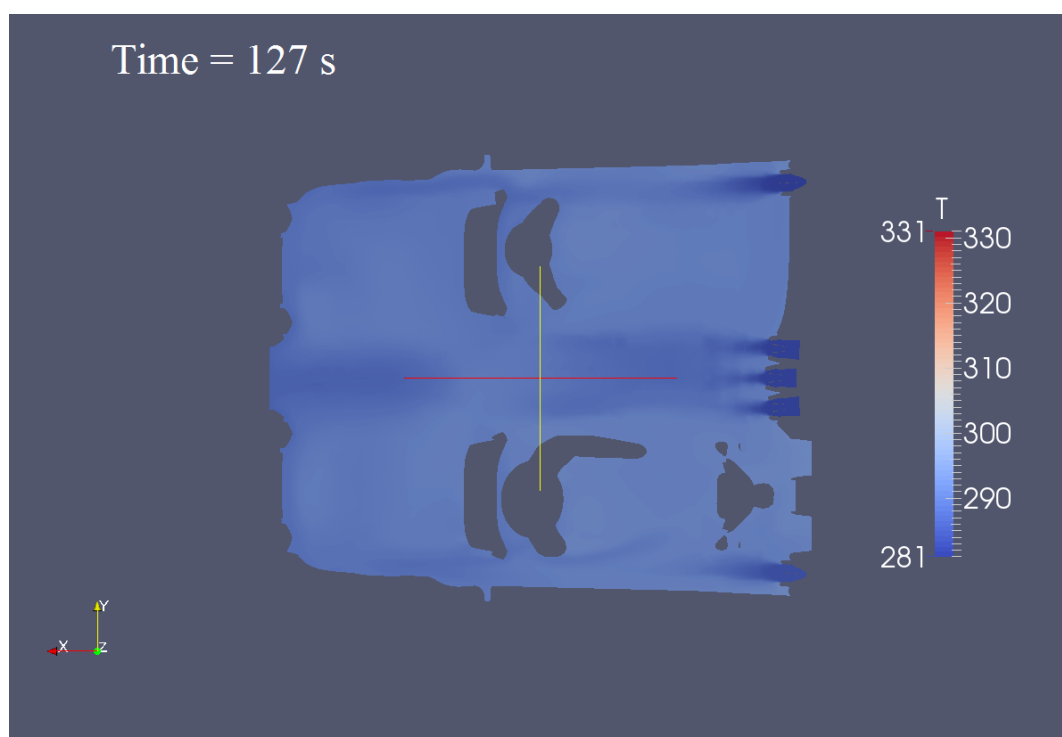


Fig.4. 41 Sezione orizzontale del campo di temperature all'altezza delle bocchette ( $t=127$ s)

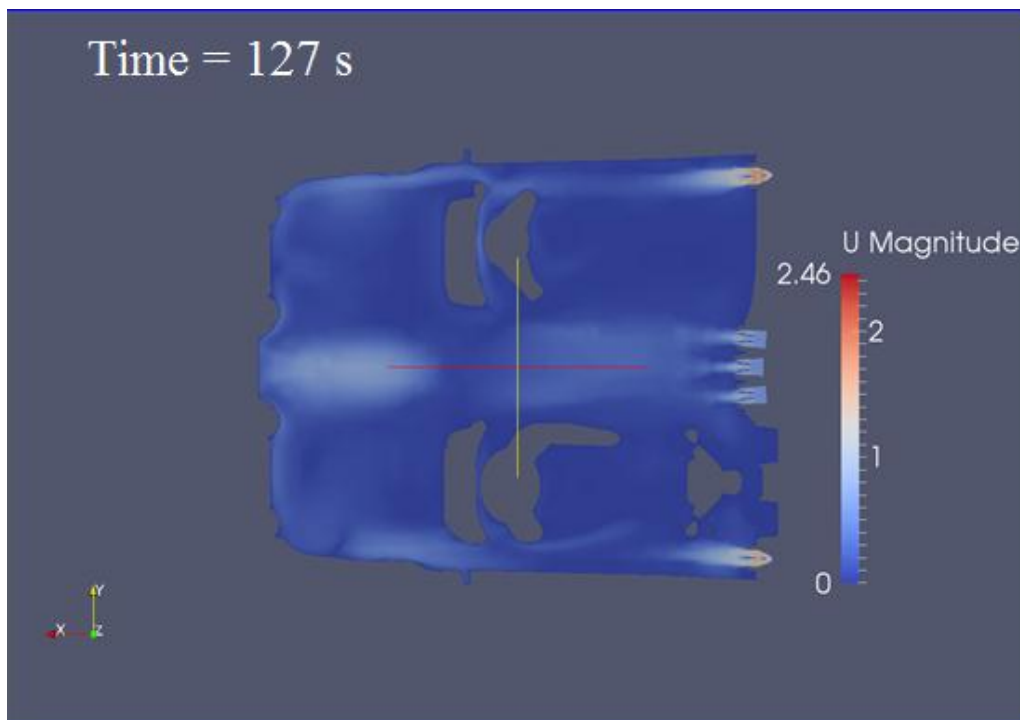


Fig.4. 42 Sezione orizzontale per il campo di velocità (modulo) all'altezza delle bocchette ( $t=127s$ )

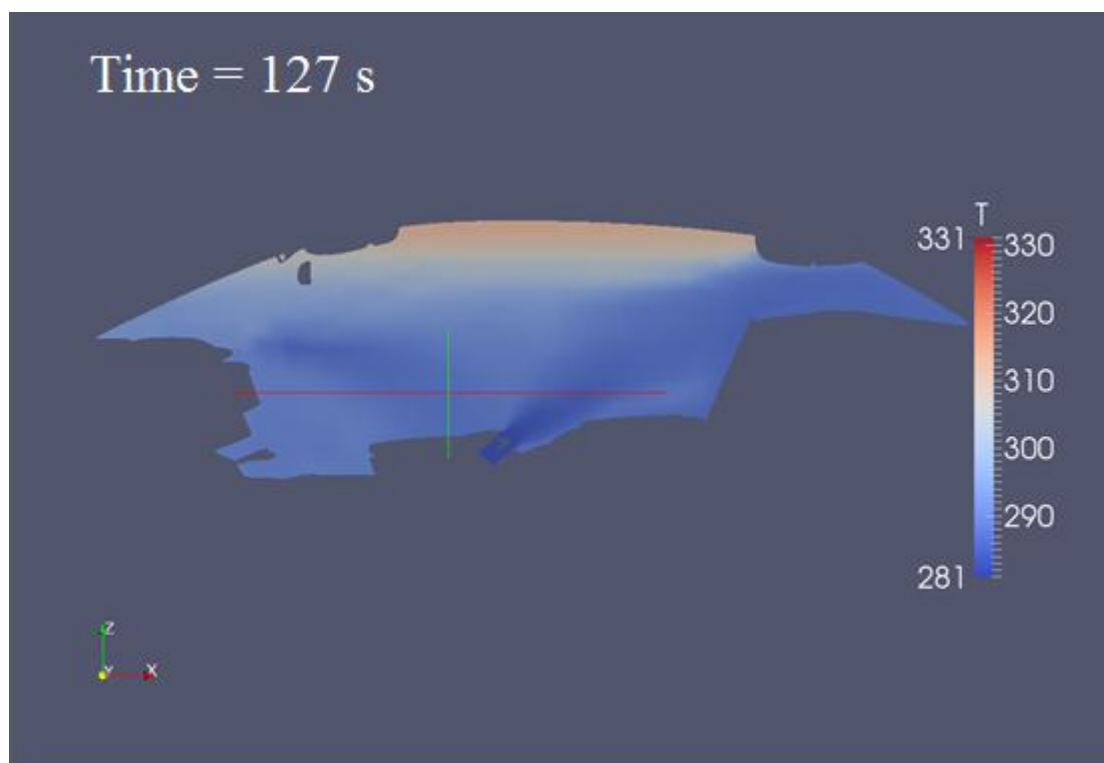
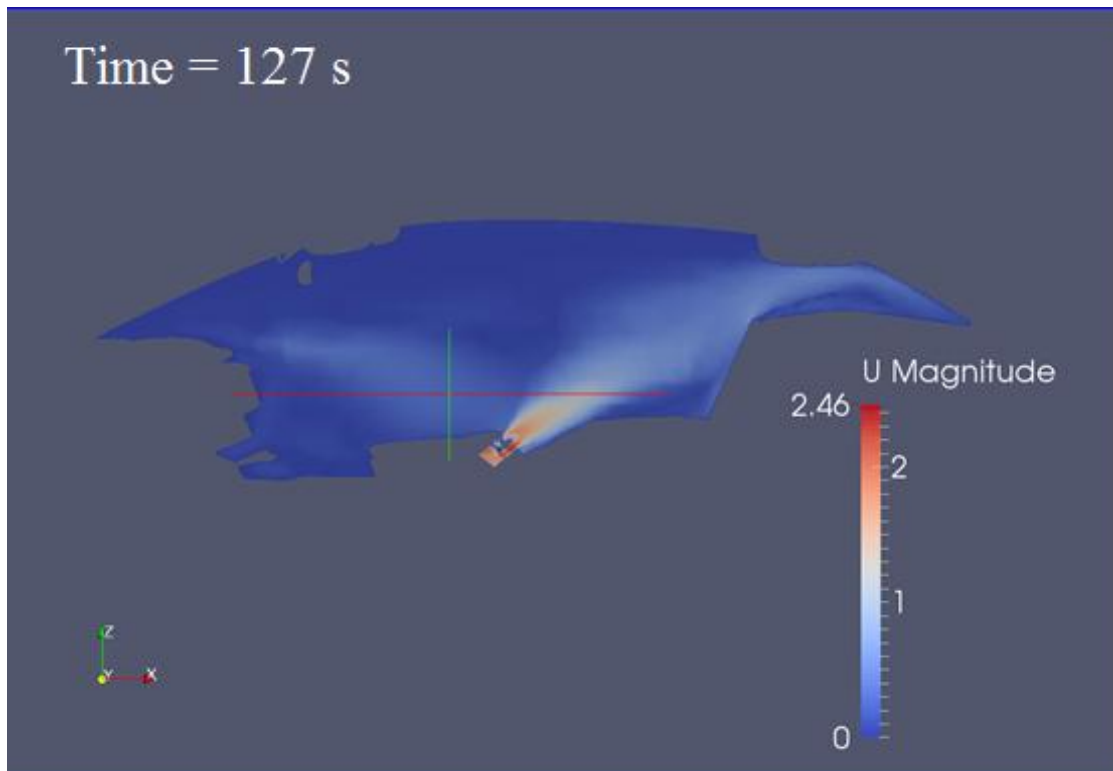


Fig.4. 43 Sezione verticale del campo di temperatura in corrispondenza delle bocchette posteriori ( $t=127s$ )



*Fig.4. 44 Sezione verticale del campo di velocità (modulo) in corrispondenza delle bocchette posteriori (t=127s)*

Dalle visualizzazioni e dai risultati delle sonde si può notare che l'abitacolo ha raggiunto mediamente una temperatura di 290 °K, ovvero 17 ° Celsius, un valore decisamente più confortevole per i passeggeri. Si nota invece, come ci si doveva aspettare, che le ultime zone a raffreddarsi sono il soffitto dell'abitacolo, questo perché l'aria fredda è più pesante e tende a stare in basso. Dalle sonde si può notare che le teste dei manichini sono meno fredde del resto del corpo. Si mostrano in Fig.4.45 le temperature sui manichini dopo 127 secondi di simulazione.



*Fig.4. 45 Temperature sui manichini dopo 127 secondi di simulazione*

Ovviamente non si è raggiunta una situazione di regime. Però si può affermare che le temperature hanno già raggiunto dopo 127 secondi una condizione accettabile ai fini del comfort degli occupanti. I risultati sono stati ottenuti usando delle approssimazioni, perciò vanno usati con dovute attenzioni.



# Conclusioni

Al termine di questo lavoro si possono trarre le seguenti conclusioni:

- La disposizione delle bocchette, della potenza dell'impianto di condizionamento in dotazione alla F151, si può dire, è molto efficace. Ovviamente ciò va detto con le dovute approssimazioni, nella realtà fisica i tempi di raffreddamento saranno molto maggiori per dei fattori che non abbiamo potuto tenere conto: è stato trascurato l'effetto serra, per i motivi citati nel Capitolo 3, e perciò le temperature effettive all'interno dell'abitacolo saranno sicuramente maggiori, e ciò sia in partenza che durante il transitorio; nella simulazione l'aria all'interno delle bocchette viene immessa istantaneamente alla temperatura più bassa e ciò non può essere reale in quanto l'impianto di condizionamento ci metterà del tempo a raggiungere la temperatura di regime impostata. Per simulare questo effetto potrebbe essere uno spunto per futuri studi implementare un flusso d'aria con temperatura variabile nel tempo (non implementato in Helyx). Fatte queste premesse si può comunque affermare che il comfort termico viene raggiunto in tempi bassi.
- Helyx è un software abbastanza giovane e semplice da usare, e sicuramente ha molto potenziale, ma attualmente non risulta ancora paragonabile a programmi come Fluent in termini di modelli fisici e customizzazione dei parametri. Il meshatore implementato in Helyx è efficace in quanto permette effettivamente di risparmiare del tempo nel pulire le geometrie ma in alcuni casi necessita comunque l'intervento di altri software per avere delle corrette mesh. Infine il solutore non stazionario è risultato poco parametrizzabile e non c'è stato modo di ridurre i tempi computazionali. In questo senso dovrebbero andare i miglioramenti nelle prossime versioni.

# Bibliografia

- I. *Lombardi G.: Lezioni di Aerodinamica degli aeromobili, dispense*, Università di Pisa, Pisa, Italia
- II. *America Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), Handbook of Fundamentals, Inch-Pound Edition*, Atalanta, USA, 2009
- III. *Engys Ltd., Helyx Kernel User Reference Guide*, Londra, UK, 2013
- IV. *Kartsiotis S., Comfort termico nel settore automobilistico: Validazione di un software CFD basato su tecnologia OpenFoam*, Tesi di Laurea specialistica, Università di Pisa, Pisa, Italia, 2014
- V. *D'agostino G., Flusso non stazionario su una vettura da competizione ad elevate prestazioni*, Tesi di Laurea specialistica, Università di Pisa, Pisa, Italia, 2014
- VI. *Agresta G., Comfort termico nel settore automobilistico: validazione di un modello di comfort*, Tesi di Laurea specialistica, Università di Pisa, Pisa, Italia 2009
- VII. *Lombardi G., Maganzi M., Comfort termico in abitacolo: confronto tra F131 e F151*, Pisa, Italia, 2009
- VIII. *Ansa©, v.13.1.2, User's Manual*, Beta-CAE System S.A
- IX. *Helyx©, v.2.1.1, User's Manual*, Engys Ltd.
- X. <http://www.openfoam.com>
- XI. <http://www.paraview.org>
- XII: <http://engys.com/it>